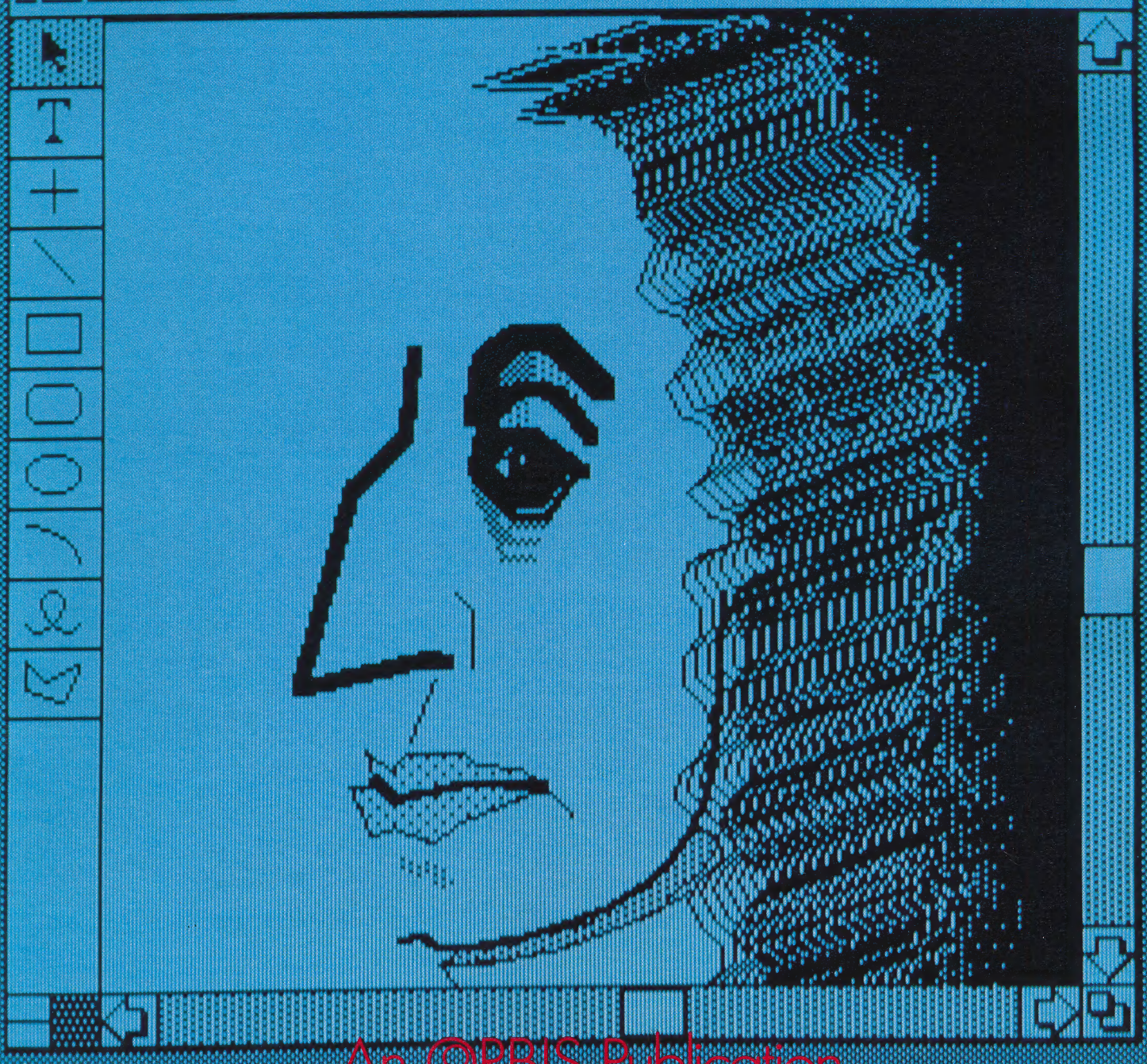
ISSN 0265-7015

80p 50

THE HOME COMPUTER ADVANCED COURSE

MAKING THE MOST OF YOUR MICRO

Blaise Pascal 1626-1662



An ©RBIS Publication

IR £1 Aus $1.95 NZ $2.25 SA R1.95 Sing $4.50 USA & Can $1.95

CONTENTS

APPLICATION

OF SIGS AND SYSOPS Our series on computer communications ends with a look at the variety of activities made possible by bulletin boards — 981

HARDWARE

TO BOLDLY GO It's almost two years since the Enterprise Sixty Four was first announced. We assess whether the long wait has been worthwhile — 989

SOFTWARE

THE CRUCIAL PATH MacProject is a planning program for the Apple Macintosh that embodies a concept known as 'critical path analysis' — 984

WHO YOU GONNA CALL? Ghostbusters: you've seen the movie and hummed the tune; now play the game . . . — 1000

COMPUTER SCIENCE

A STRICT DISCIPLINARIAN We present our introduction to PASCAL, a language designed to improve programming technique and do away with lengthy debugging sessions — 986

JARGON

FROM LOGIC GATE TO LOGGING ON A weekly glossary of computing terms — 988

PROGRAMMING PROJECTS

THE HAUNTED FOREST We round up our adventure game tutorial series with a full listing of the program we have constructed — 997

MACHINE CODE

ON FILE The BBC Micro's OS calls to the filing system are this week's subject — 994

WORKSHOP

A PLAN TO MODIFY We take a look at the principles behind the construction of an interface for the Spectrum that will allow it to control the Workshop robot — 992

JOYSTICK JURY Bonus extra software reviews. This week we look at Zombie Zombie and Eddie Kidd Jump Challenge — INSIDE BACK COVER

Next Week

- **The Apricot Portable has an 80 by 25 LCD screen, infra-red transmission and speech recognition. We look at the advanced features of this machine.**
- **We begin a new series about computers and education. Our introductory feature discusses how the micro has quickly become an important teaching tool.**
- **We publish the full listing for our Digitaya adventure program.**

QUIZ

1) Which control line do we use to send a signal to an external device from a Sinclair Spectrum?
2) How many key words are there in PASCAL's vocabulary?
3) What are the names of the three parts of a typical transistor?
4) What are the custom-built video and audio chips on the Enterpise Sixty Four called?

Answers To Last Week's Quiz

1) Because the Commodore 64 has only three voices, it cannot play polyphonically with the rhythm playing at the same time.
2) It is probably accurate. However, the 128 Kbytes is Read Only Memory, none of it is set aside for the user.
3) A bus network does not require the information to pass through a third computer. Generally, ring and star networks do.
4) A 'parameter block' is an area of memory set aside to pass parameters over to the operating system from OSWORD calls.

Managing Editor Mike Wesley; **Editor** Stephen Cooke; **Art Editor** Claudia Zeff; **Production Editor** Bobby Pickering; **Technical Editor** Steve Colwill; **Designer** Julian Dorr; **Art Assistant** Liz Dixon; **Staff Writer** Stephen Malone; **Sub Editor** Jonathan Kaye; **Contributors** Geoff Bains, Nick Walsh, Joe Pritchard, Steve Malone, Karl Dallas, Surya, Steve Colwill; **Software Consultants** Pilot Software City; **Group Art Director** Perry Neville; **Managing Director** Stephen England; **Published by Orbis Publishing Ltd: Editorial Director** Brian Innes; **Project Development** Peter Brookesmith; **Executive Editor** Maurice Geller; **Production Manager** Peter Taylor-Medhurst; **Subscription Manager** Christine Allen; **Designed and produced by Bunch Partworks Ltd; Editorial Office** 14 Rathbone Place, London W1P 1DE; **© APSIF Copenhagen 1984; © Orbis Publishing Ltd 1984: Typeset by Universe; Reproduction by Mullis Morgan Ltd; Printed in Great Britain by Heanor Gate Printing Ltd, Derby**

HOW TO OBTAIN ISSUES AND BINDERS FOR THE HOME COMPUTER ADVANCED COURSE – Issues can be obtained by placing an order with your newsagent or direct from our subscription department. If you have any difficulty obtaining any back issues from your newsagent, please write to us stating the issue(s) required and enclosing a cheque for the cover price of the issue(s). **AUSTRALIA** – please write to: Gordon & Gotch (Aus) Ltd, 114 William Street, PO Box 767G, Melbourne, Victoria 3001. **MALTA, NEW ZEALAND & SOUTH AFRICA** – Back numbers are available at cover price from your newsagent. In case of difficulty, write to the address given for binders. **UK/EIRE** – Price: 80p/IR£1. Subscription: 6 months: £23.92. 1 Year: £47.84. Binder: please send £3.95 per binder, or take advantage of our special offer in early issues. **EUROPE** – Price: 80p. Subscription: 6 months air: £43.68. Surface: £34.84. 1 year air: £87.36. Surface: £69.68. Binder: £5.00. Airmail: £8.25. **MALTA** – Obtain binders from your newsagent or Miller (Malta) Ltd, MA Vassalli Street, Valetta, Malta. Price: £3.95. **MIDDLE EAST** – Price: 80p. Subscription: 6 months air: £45.76. Surface: £34.84. 1 year air: £91.52. Surface: £69.68. Binder: £5.00. Airmail: £8.25. **AMERICAS/ASIA/AFRICA** – Price: US/CAN$1.95/80p. Subscription: 6 months air: £54.08. Surface: £34.84. 1 year air: £108.16. Surface: £69.68. Binder: £5.00. Airmail: £9.50. **SOUTH AFRICA** – Price: SA R1.95. Obtain binders from any branch of Central News Agency or Intermag, PO Box 57394, Springfield 2137. **SINGAPORE** – Price: Sing $4.50. Obtain binders from MPH Distributors, 601 Sims Drive, 03-07-21, Singapore 1438. **AUSTRALASIA/FAR EAST** – Price: 80p. Subscription: 6 months air: £58.24. Surface: £34.84. 1 year air £116.48. Surface: £69.68. Binder: £5.00. Airmail: £9.75. **AUSTRALIA** – Price: Aus$1.95. Obtain binders from First Post Pty Ltd, 23 Chandos Street, St Leonards, NSW 2065. **NEW ZEALAND** – Price: NZ$2.25. Obtain binders from your newsagent or Gordon & Gotch (NZ) Ltd, PO Box 1595, Wellington.
ADDRESS FOR BINDERS AND BACK ISSUES – Orbis Publishing Limited, Orbis House, Bedfordbury, London WC2 4BT. Telephone 01-379 5211. Cheques/postal orders should be made payable to Orbis Publishing Limited. Binder prices include postage and packing and prices are in stirling. Back issues are sold at the cover price, and we do not charge carriage in the UK.
NOTE – Binders and back issues are obtainable subject to availability of stocks. Whilst every attempt is made to keep the price of the issues and binders constant, the publishers reserve the right to increase the stated prices at any time when circumstances dictate. Binders depicted in this publication are those produced for the UK and Australian markets only. Binders and Issues may be subject to import duty and/or local taxes, which are not included in the above prices unless stated.
ADDRESS FOR SUBSCRIPTIONS – Orbis Publishing Limited, Hurst Farm, Baydon Road, Lambourn Woodlands, Newbury Berks, RG16 7TW. Telephone: 0488-72666. All cheques/postal orders should be made payable to Orbis Publishing Limited, Postage and packaging is included in subscription rates, and prices are given in sterling.

COVER GRAPHICS AND PHOTOGRAPHY BY IAN McKINNELL ON THE MACINTOSH

OF SIGS AND SYSOPS

HCAC BULLETIN BOARD
Notice To All Members:
This Bulletin Board is beginning a new service for members - a lonely hearts personal message board, as a subset of (M)essaging. These messages are open to be read by all users, and are intended to provide a way for all you crazies to meet someone on your wavelength. Respond to lonely hearts messages via E-mail.
Message Area Menu
(C)heck (S)can
(R)ead
(M)ail (T)ime
(L)onely Hearts
(E)xit
Ye Magick Toy-Shoppe.
This Toy-Shoppe is in a maze
If you haven't time to explore leave it now.
(Q)uit
(W)est (N)orth (S)outh
Command: s
The Map Room
Ah, I wondered when you would come in here.
I can't find the map so you will have to look for it.
(N)orth
(W)est (S)outh (E)ast
Command: s
Murphy & Sons
This room is being rebuilt
Beware of falling Masonry.
(N)orth
(W)est (S)outh (E)ast
Command: s
8395
From: Liz
Female Trekkie with TRS-80 would like to correspond with articulate, attractive male into disk-based adventures and Traveller or Superheroes. Have you seen Gremlins yet? End.
8775
From: Max
Male Commodore 64 user, glasses (owlish appearance and wisdom), 23 would like to meet quiet, studious female. End.
8912
From: George
Can you (female, 20-25) tear me (male, 27) away from my computer for a few hours? I lay down the gauntlet to any and all. End
COBOL ANALYST/PROGRAMMER
David Steel opens store
High School girls win prize
From: MIKE
To : ERIC
Clever move, but you forgot about my Knight. Next move is
From: ERIC
To : MIKE
NO, I DIDN'T. NEXT MOVE IS
25... R-K N c h
From: MIKE
To : ERIC
@#$?!!!!
26. Resign
Give me a day to recover and we'll start a new game
JANET Phone your Mum
HI-FI FOR SALE
PHIL 3:00 PHONE SON
Commodore 64

STEVE CROSS

Bulletin boards, established by micro user groups or enterprising individuals, offer the hobbyist with micro and modem a simple and informal exchange point to communicate with fellow enthusiasts.

The bulletin board has its origin in a US computer club. The club decided that the pin-up notice board it was using to display club bulletins wasn't in keeping with its hi-tech outlook! A few members decided that it wouldn't be difficult to design a dial-up bulletin system on which the club could display information for members. Other facilities that seemed useful were for individual members to exchange both public and private messages, and the up-loading and down-loading of programs from the club's software library. The bulletin board was born.

Today there are hundreds of computer bulletin boards worldwide. A few are run by clubs and user groups, but most are run by individual hobbyists. Almost all of them are open to anyone who wants to use them and access is free of charge. You simply dial up, enter your name and answer a few questions about your terminal.

There are several kinds of bulletin board systems. The two most common are The Bread Board System (TBBS), so-called because of the ease with which it can be customised, and the Central Bulletin Board System (CBBS). CBBS systems exchange lists of users, so if you register with one board you will automatically be registered with other CBBS boards. There are also a number of boards running on home-devised software. The most popular system in the UK is the TBBS board. At present 10 of them are in operation. TBBS systems are divided into four main areas: general messaging, special interest groups (SIGs), electronic mail and file transfer.

Any Messages?
Computer bulletin boards offer a variety of services to users, including personal or general messages, product and technical information and games. Their main purpose is to provide a forum for users to communicate with others with similar interests

The general messaging facility is for public messages not related to a specific subject or machine. Messages left in this area include requests for technical help (and — all being well — answers from other users), jokes, comments on new machines and modems, hints and tips, club news, reports on new books and magazines, and general chat.

Special interest groups give public messages about specific machines and well-defined areas of interest. Typical SIGs are BBC, Tandy, Commodore, CP/M, Modems, and Sales and Wants. The SIGs offered vary from board to board, and if you'd like to see a new SIG established, leave a message with the 'sysop' (system operator) asking him or her to start one. SIGs operate on demand; if they attract enough interest, the sysop will run them.

Electronic mail, normally referred to simply as E-mail, allows for the transmission of confidential messages from one user to another. E-mailed messages can be read only by the author, the recipient and the sysop, confidentiality being maintained by passwords. When you register, you are normally asked to choose a password. This is stored against your name, and you are asked for this password each time you log on.

The file transfer area enables users to exchange public-domain software. It is designed for people who have written a BASIC program and want to share it with other users. You don't get paid for uploading your own programs and there's no charge for downloading other people's contributions. As with SIGs, this area is normally divided by machine type. Thus, you will be asked which machine you have and then offered a menu of programs for that machine.

Other facilities vary from board to board — for example, TBBS London runs an electronic game of Diplomacy; TBBS Nottingham has a real-time adventure game; and Mailbox-80 Liverpool has a version of the artificial intelligence game, Eliza. Most boards have an introductory message for new users, which the initiate is given automatically when first logging on; additional information is often contained in<I> (information) files.

HCAC Communications Datasheet	
Bulletin Boards	
Telephone Number (Data)	See list
Telephone Number (Enquiries)	N/A
Cost	Free
Operating Times	Vary
Baud Rate(s)	300/300 (a few 1200/75 also)
Parity	None
Data Bits	8
Stop Bits	1
To Get Help	Usually <I>nfo
To Log Off	Usually <G>oodbye
Electronic Mail?	Yes
Public Messaging?	Yes
Commercial Software?	No
Free Software?	Yes

Selected Bulletin Boards

Bulletin boards tend to come and go, so it would be pointless to present a comprehensive list. Those included here, however, are established and are likely to be around for some time. You'll find lists of other boards on these systems

Board	Hours	Phone
TBBS London	24	01-348 9400
Mailbox-80 Liverpool	24	051-428 8924
The Blandford Board	24	0258-54494
CBBS Surrey	24	04826-25174

Logging On

Once you're on a board you should find most things self-explanatory, but here are a few hints to help you out:

- Check the operating times carefully. The main boards are accessible 24 hours a day, but others are available for only a limited period. If you're in any doubt, call during daytime hours. Some boards operate on a 'ring-back' basis: let the phone ring once, replace the receiver, then re-dial. The first ring lets the modem know that you want to talk to it rather than its owner.
- Make sure your terminal is ready for use before dialling. Most systems will automatically hang up if they don't receive a carrier tone from you within about five seconds of answering. Some boards take longer to answer than others, and if the previous caller has only just rung off, the system may take up to three minutes to recycle ready for the next caller.
- Be prepared to hear a human voice — even 24-hour boards need daily maintenance. It's very irritating for the sysop to answer the phone and have the caller hang up after expecting a carrier. Simply ask whether the board is available.
- If the number is engaged, keep trying. Most callers are restricted to between 15 and 30 minutes per call, and the busiest boards have only a few minutes between calls. An auto-dial phone or modem is a very useful asset in this situation.
- When you log on, don't use a false name without informing the sysop. Don't leave messages that some users may find offensive; if you do, you're liable to find, next time you call, that the sysop has deleted your name from the user log. As a discouragement to pranksters, many systems don't allow you full use of the board on your first call (you may have read only access, for example). You will normally have better access by the next day if you use the <N> (new user) or <R> (registration) facility.

Sample Session On A Bulletin Board

Welcome to the Blandford Board

Open 24 Hours Daily

When you get a carrier signal press <CR> once or twice. You will then get the sign on message:

CCITT at 300 Baud
8 Bits - No parity - 1 Stop

Member of A.F.P.A.S.

You should find the menus easy to follow. If you're not sure what something means, try it and see!

First Name? Home Computer
Last Name? Advanced Course
Calling from (Home Town)? London

Some boards give a terse sign on message, others reams of notes. Pressing 's' normally skips unwanted text. You are then asked to log on by entering your first and last names, and home town

TBBS Welcomes HOME COMPUTER
ADVANCED COURSE
Calling from LONDON
Is this correct y

The board checks this information with you, in case two people with the same name are using it

OK

Terminal Width (10-132)? 52

<A> TRS-100 <B> TRS-1/3/4 <C> APPLE
<D> ATARI <E> IBM-PC <F> OSBORNE
<G> BBC <H> APRICOT <I> COMMODORE 64

You are then asked about your terminal. Apart from screen width, the answers given here will suit most micros:

Enter letter of your terminal, or <CR>: <CR>

Can your terminal print lower case? y
Does your terminal need line feeds ? y
How many Nulls (1-50)? 0

If your terminal is not listed, just press <CR>:

The Blandford Board

<N>ew user registration
<B>ulletins
<M>essaging section
<U>tility area
<I>nformation
<C>hat with Sysop
<G>oodbye to terminal

You are then asked to choose a password for future log ons (to prevent other people reading messages addressed to you) and you may be given information about the system. You are told whether there are any messages for you, and presented with the opening menu

Command: m

...And so on. You are then shown all messages with 'Commodore' in the heading. To exit from this, press s for 'stop'. Together with p to pause and <CR> to restart this works in many parts of the board.

Messaging Section

<O>pen Message area
<M>ail - Private and Confidential
<S>IGs for Special Interests
<B>landford Computers (Os House)
<U>tility area
<E>xit messaging section

On most boards you have to register before being allowed full use of the system. It is quite common to be given read-only access on your first visit. Once you've done this, go to the messaging menu:

Command: o

Open Message Area

<Q>uick-scan Open message headings
<R>ead Open messages
<L>eave an Open message

Here is a typical list of special interest groups (SIGs):

<M>ail - Private and Confidential
<E>xit Open message area
<G>oodbye to terminate

Command: r

Type P to pause, S to stop, N to skip to next msg

<F>orward or <R>everse Multiple
<N>ew Messages
<M>arked Messages
<S>elective Retrieval
<I>ndividual Message(s)
<A>bort Retrieve

Which One? s

Select <F>rom, <T>o, or <S>ubject? s
Enter text to select by: Commodore

Reply to msg option (Y/N)? y

Msg£: 3330 *OPEN*
07/11/84 20:07:09 (Read 14 Times)
From: NORM PHILLIPS
To: ANY COMMODORE 64 USER
Subj: COMMODORE GEN: AS A NEW USER OF THIS SYSTEM I WOULD BE GRATEFUL FOR ANY HELP ANYONE COULD OFFER ME TO GET THE BEST FROM THE SYSTEM WITH MY COMMODORE. I WOULD ALSO BE INTERESTED IF THERE ARE ANY USERS IN MY LOCAL AREA AND IF SO COULD THEY CONTACT ME?

<N>ext msg, <R>eply, or <S>top? R

You are not authorised to write messages on this board.

Msg. has replies, read now (Y/N)? n

At the start of each call you are set for access to all combined boards. You may toggle a board's activation state by pressing the letter indicated in the menu below. Any boards not listed as active will not display messages to you.

Active combined boards: Sales & Wants, TRS-80, BBC, APPLE, MODEMS, ATARI, C'64
<A> Sales & Wants <B> TRS-80
<C> BBC <D> APPLE
<E> MODEMS <F> ATARI
<G> C'64 <CR> to Exit

Which One?

IAN McKINNELL ON THE MACINTOSH

THE CRITICAL PATH

When planning a complex project and overseeing the various tasks that need to be done, there are obvious advantages in maintaining a visual record of progress such as a time chart. MacProject is a program, unique to the Apple Macintosh, that enables even those without graphic gifts to set up a neat and accessible project plan.

Critical path analysis (CPA) is a powerful project management tool in which, according to the *Longman Dictionary of Business English* (1982), a project 'is divided into many separate steps, each of which is represented on a chart by an arrow having a time value. The result is a series of parallel chains which come together at certain points. The chain taking the longest time is the 'critical path', because it determines the total length of the work. The chart shows the points at which various materials, tools, etc. will be needed, just when each step must be begun and completed, and by whom'.

Attempts have been made to produce CPA programs for computers, but since a CPA network is basically a graphics concept, the complexities of drawing and redrawing a network each time new information is input have defeated the programmer, and the programs tend to be text-oriented and rather unfriendly. This is counter-productive: the whole point of a CPA network is to represent a complex sequence of actions in a form that is immediately accessible. Until now, computer-based CPA programs have tended to complicate rather than simplify the issue.

The graphics-based Apple Macintosh is ideally suited for this kind of work, and MacProject makes the creation of a network (or project chart, as the program calls it) so easy that it really does simplify the management of a project. Dates and times can be changed at will and this results in an instant recalculation of total project times; new tasks can be inserted and redundant tasks deleted; and resources allocated, controlled and reallocated — all in no more time than it takes to point the Macintosh 'mouse' and wait for the Mac drive to overlay the necessary routine from disk.

The manual, friendly but concise, draws a simple example of a family barbecue to illustrate the basic concept, breaking down the project into five tasks:

MAKE A SALAD
SET THE TABLE
START THE FIRE
COOK THE RICE
BROIL THE STEAKS

and leading to completion of the project:

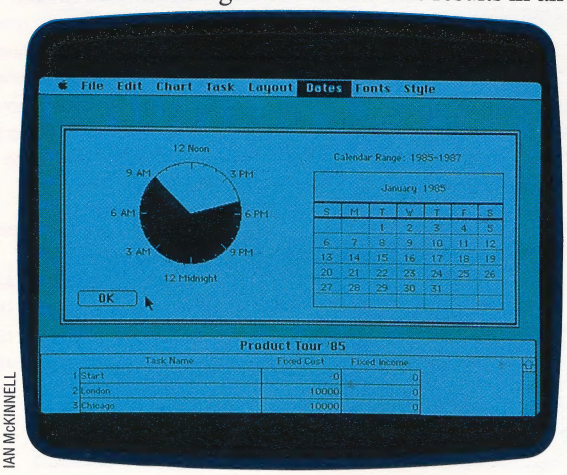
EAT DINNER

Most of the tasks can be conducted concurrently, but it is impossible to broil (grill) the steaks until the fire has been lit, so the critical path goes from 'light fire' via 'broil steaks' to 'eat dinner'. If the fire is hard to get going, then the dinner will be delayed, even if all the other tasks have been completed on time. Similarly, if the person appointed to make the salad turns up late, the salad-making may become critical, and the hungry diners may have to wait around while the salad is tossed (and the steaks cool, burn or just spoil).

As the manual admits, you don't need a £100 program to plan a barbeque, but the same principles can be applied to any project, from organising a business conference to the construction of an oil refinery.

Chart task boxes are drawn by moving the mouse. When setting up of the chart is begun, these are dated starting from January 2 but it is possible to change the dates, either by choosing a new start date or, if a deadline is being worked towards, by choosing a finish date. Once the first box has been created, further boxes of the same size can be linked to it by moving the mouse from inside the previous box to the required position. Clicking the mouse inside a box allows task labels to be typed in; the program automatically centres them within the available space. Boxes can be enlarged by clicking on the border, which then displays eight 'handles'. Moving the cursor to one of these and shifting the mouse will change the size. By moving the cursor to the border but not to a 'handle', and moving the mouse, the box can be moved on the chart.

To ensure a tidy, squared-up project chart, INVISIBLE GRID should be selected from the LAYOUT



IAN McKINNELL

Screen Options
The MacProject calendar, shown here as the top window on a Macintosh screen, allows you to enter the number of working hours and working days allowed for completion of the project, as well as taking into account holidays, weekends and other special considerations. Below this is the Cash Flow sheet, which keeps a running count of expenses for the project

menu option before the first task box is drawn. All boxes will then be aligned accurately.

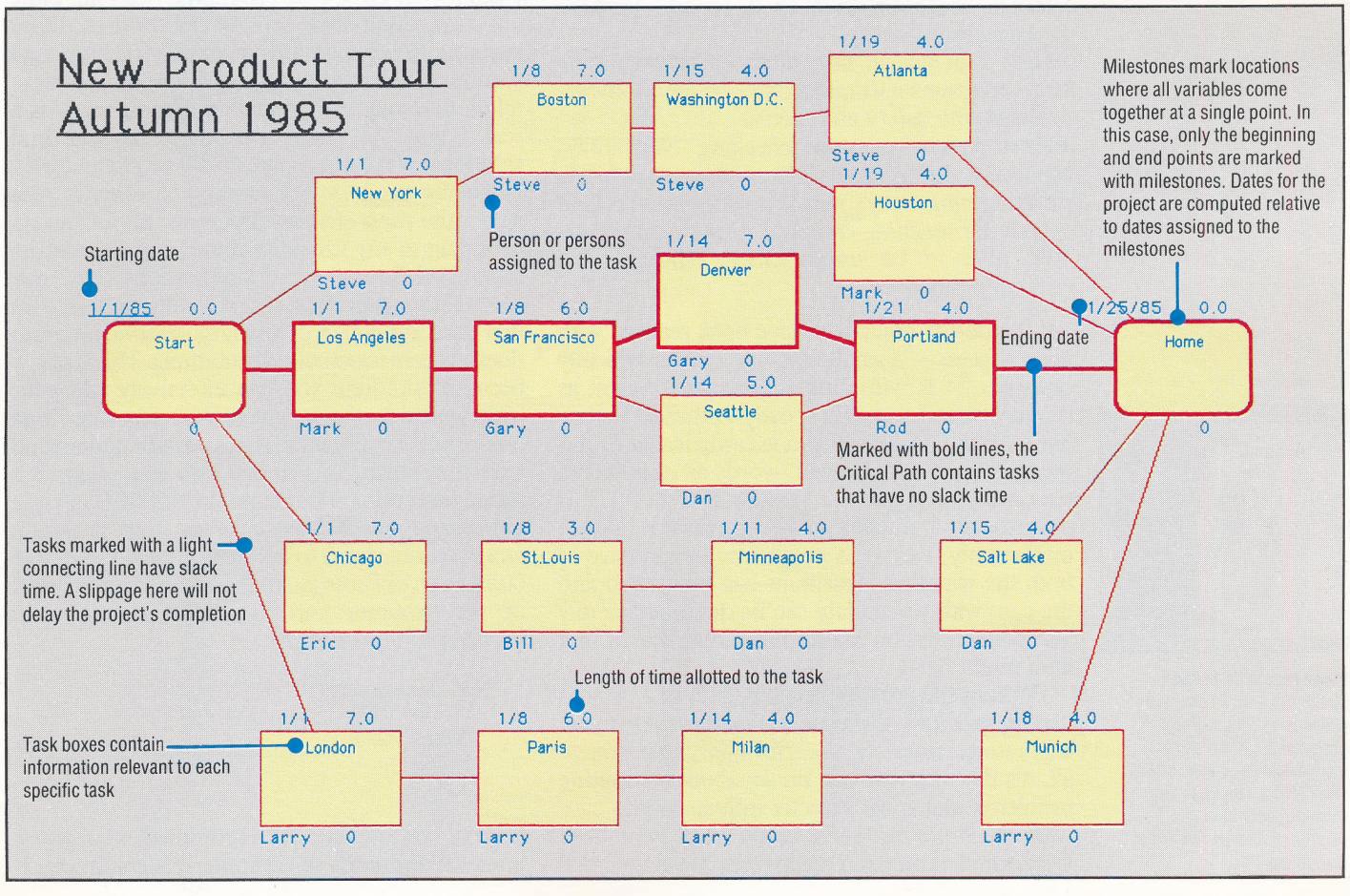
When a box has been labelled, it is possible to input TASK INFO from the appropriate menu item. This allows for the input of time (in days, minutes, weeks or whatever) and resources, which may be people or supplies. It is also possible to display on the main chart up to four items of information around each task box: Resource or Earliest Start at top left; Duration, Resource or Earliest Finish at top right; Latest Start, Fixed Cost or Resource at bottom left; and Latest Finish, Fixed Income or Resource at bottom right. When the procedure chart has been completed, the critical path is indicated with a bold rule, while non-critical paths are indicated with a light rule.

Meanwhile, a series of analysis charts has been prepared without any further action having been taken by the user. RESOURCE TIMELINE displays a bar chart showing how resources are allocated. TASK TIMELINE is similar but lists the tasks rather than the resources (or staff) allocated to each. 'Slack time' (the difference between earliest and latest finish time, during which the task can be completed without delaying the total project) is shown in grey. PROJECT TABLE produces a table of all tasks, showing the critical tasks in bold and displaying the number of days allocated, earliest start and finish, latest start and finish, fixed cost, resource cost, fixed income and resource names for each. Three tables relate to costs, the purposes of each being self-explanatory: TASK COST ENTRY, RESOURCE COST ENTRY, and CASH FLOW TABLE.

From the LAYOUT menu option it is possible to show a schematic diagram of the entire chart (without explanatory text) or to change the chart size. If a number of boxes at the end of a chart has been deleted, the empty space is still shown and this option allows the chart to be reduced accordingly.

MacProject is an effective implementation of the principles of Critical Path Analysis, which are explained to the lay reader in clear, easy-to-understand English without the use of jargon (CPA is not even mentioned). The graphics capabilities of the Macintosh are used to the full and in such a way that it is difficult to imagine a similar implementation becoming available on any other machine. It is unlikely, of course, that a user would be swayed by the sophistication of the software to purchase the computer for this use alone, but MacProject is nevertheless a particularly good example of the way in which this graphics-oriented machine can assist the non-graphically minded.

MacProject: For the Apple Macintosh
Price: £100 (plus VAT)
Distributor: Apple Computer (UK), Eastman Way, Hemel Hempstead, Herts HP2 7HQ
Authors: Debra Willett and Stephen D. Young of Solosoft
Format: Disk


New Product Tour
Autumn 1985
Milestones mark locations where all variables come together at a single point. In this case, only the beginning and end points are marked with milestones. Dates for the project are computed relative to dates assigned to the milestones
Starting date
Person or persons assigned to the task
Ending date
Marked with bold lines, the Critical Path contains tasks that have no slack time
Tasks marked with a light connecting line have slack time. A slippage here will not delay the project's completion
Length of time allotted to the task
Task boxes contain information relevant to each specific task
Start
Los Angeles
San Francisco
Denver
Seattle
Portland
Home
Atlanta
Boston
Washington D.C.
New York
Houston
Chicago
St.Louis
Minneapolis
Salt Lake
London
Paris
Milan
Munich

MacSampleProject
This MacProject chart represents a tour of several cities in which a company is promoting a new product. By providing a visual schedule for the trip, MacProject speeds up and simplifies the allocation of people and time to the project

A STRICT DISCIPLINARIAN

PASCAL's Major Programming Features
- Completely free format and layout of the program text
- Flexible requirements in the naming of program objects
- The ability to define new command 'words'
- Simple and consistent syntax
- Data typing
- Modular programming structure
- Flexible control of data and processes
- Natural recursion
- Excellent compile–time error diagnosis
- A small, highly efficient compiler

We begin a series of articles about PASCAL, the most influential of the high-level programming languages. Here, we discuss the language's origins and the principles behind its development. Often characterised as a 'strict disciplinarian', PASCAL is designed to encourage good programming technique.

The PASCAL programming language was originated by Professor Niklaus Wirth of Zurich around 1970. It is named after the 17th century French mathematician and philosopher, Blaise Pascal, who invented the first four–function calculator. The PASCAL language was mainly influenced by ALGOL 60, and was a direct response by Wirth to the oversized and complex ALGOL 68. He intended that PASCAL would:

- Allow precise expression of programming concepts and structures.
- Demonstrate that a small machine–independent language with a rich and flexible set of data, statement and program structuring features could be used as a general purpose problem–solving tool.
- Help gain insight into methods of organising large programs and managing complex software projects with safety and security.
- Have extensive error checking capabilities, especially during compilation, thus minimising programming errors and providing an excellent vehicle for teaching computer programming.
- Be able to be implemented efficiently on microcomputers.

All the design objectives have been realised with great success — a small PASCAL compiler typically occupies 24 Kbytes, and is twice as efficient as FORTRAN (itself noted for speed). Although PASCAL has a small vocabulary and is easily learnt (it has only 35 'key' or 'reserved' words as opposed to over a hundred in most variations of BASIC), it is nevertheless much more powerful. More importantly, PASCAL is much more expressive in both the way that algorithms can be written and the ease with which data can be described simply and coherently, no matter how complicated that data might be.

Perhaps the greatest single benefit of PASCAL is that it provides a simple and coherent way of expressing extremely powerful algorithms. Above all, it's the *way* that one thinks about computing problems that most directly influences the ease with which we can solve them. PASCAL's natural freedom of expression means that the language is a marvellous tool for solving problems, rather than being part of the problem itself. The language has many other advantages, as well. PASCAL is a compiled language, which not only means that programs execute many times faster, but that precious memory is not used up by the presence of the source program text and a language interpreter — all you need is the compiled 'object' code.

The whole philosophy of the language is to protect you from your own folly and to prevent an error-laden program from running at all. This may seem very strange at first, particularly to BASIC programmers. Often, however, in BASIC, the quicker we get to run a program, the longer it is before we get it to work.

In fact, especially for a large program, it is actually much easier to program in PASCAL than in BASIC. Sometimes PASCAL is characterised as a 'strict disciplinarian' and, although this is often voiced as a criticism, it in fact indicates the excellence of PASCAL's error diagnosis and reporting. We all need an occasional slap on the wrists to remind us that it takes thought and care to write robust error-free programs. After all, these imposed disciplines are no more than those required to organise any program effectively and securely. The resulting reduction of the time taken to debug programs that 'almost work' is a real bonus. In short, PASCAL helps you find solutions — it's not a part of the problem itself.

To the programmer coming to PASCAL from BASIC, the most striking difference about PASCAL programs of any size is the abundance of peculiar-looking definitions and declarations that may seem rather pointless, if not totally useless. The first yard and a half of any large PASCAL program doesn't seem to actually do anything. This is partly because, although you can effectively add your own words to the language, this must be done before you can use them if PASCAL is to understand what they mean. So, whereas in BASIC a program is stated first (using statements like GOSUB 5000) and the subroutines defined *after* the main program, PASCAL enables you to define new commands like ClearScreen or Pause (so many seconds) at the start of the program and use these in the main procedure. For example:

```
begin
   ClearScreen;
   Write ('Hello!');
   Pause (3);
   ...etc.
```

PASCAL encourages a methodical, but very practical approach to computer programming. It

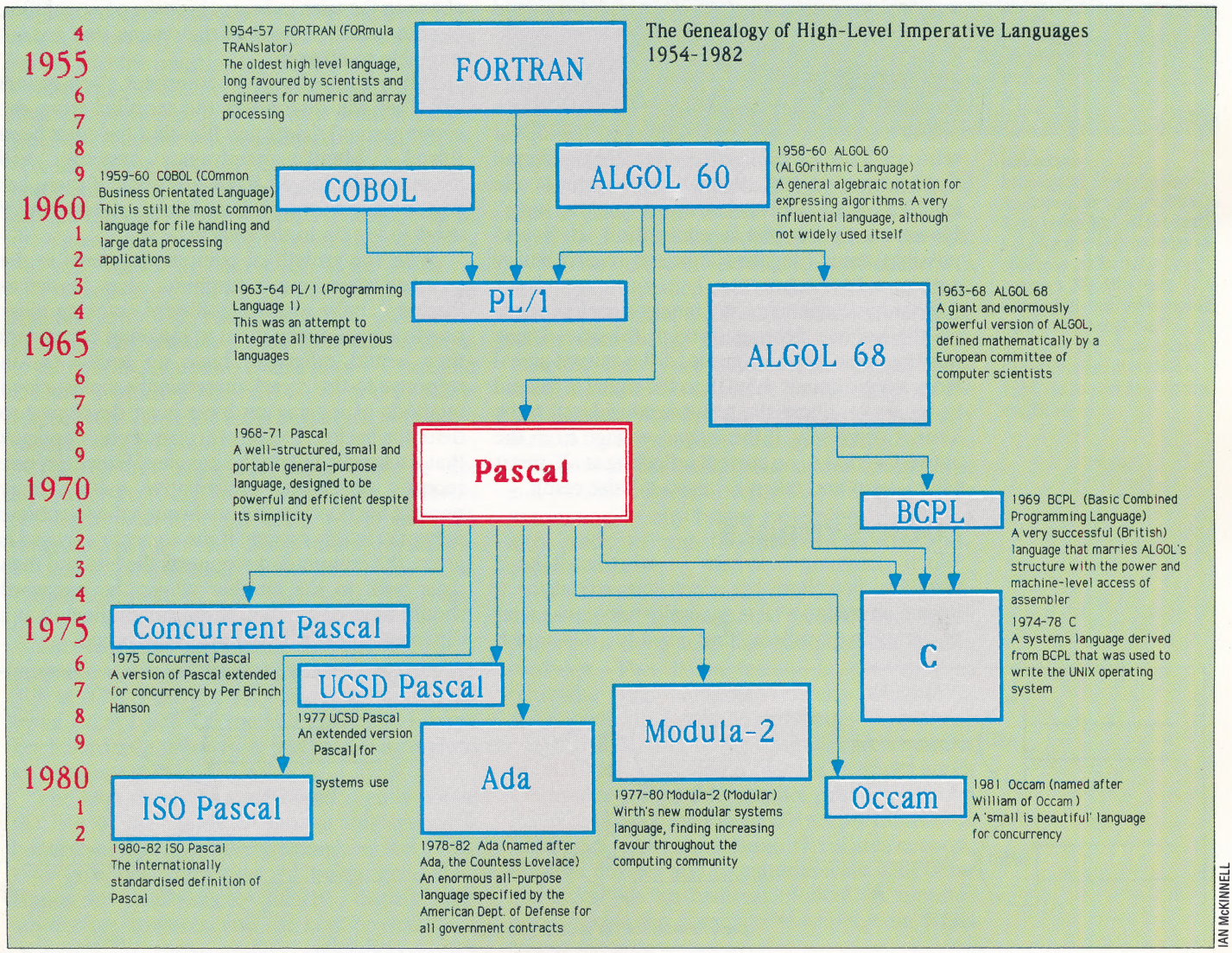
Blaise Pascal, 1623-1662
Mathematician, scientist and philosopher, Blaise Pascal designed the world's first mechanical calculator in 1642. The PASCAL language was named in honour of this contribution to the science of computing

presents the user with a very high-level conceptualised view of a computer system, so that data and processes can be defined and expressed in a natural, logical way. This ensures a high degree of inherent portability and safety, and both the error detection and diagnostics are excellent. In general, PASCAL has had a greater influence on other computer languages — and software design — than any other programming language.

Our diagram showing the genealogy of high-level languages features only the main influences of most of the important compiled imperative languages, and so does not include LISP, PROLOG or any other functional language. Nor is FORTH shown — it's almost unclassifiable! The mainstream of influence starts with ALGOL 60, and there is hardly a single modern language of any note that has not been derived, either directly or indirectly, from PASCAL. This in turn means that a sound knowledge of PASCAL will be an enormous advantage in understanding the languages of the next decade — such as MODULA-2, OCCAM or ADA.

Besides being used extensively for teaching purposes, PASCAL has been widely adopted for many commercial and systems applications. It has been used for writing software as diverse as financial packages and language compilers. The 'p-system', the famous portable operating system designed at UCSD (University of California at San Diego) in the late 1970s, was both developed and written in PASCAL. The software for Apple's Lisa and Macintosh, including their operating systems, was mainly written in either PASCAL or its derivative, CLASCAL. Many thousands of professional programmers were weaned on PASCAL, either on the University's DEC or VAX, or possibly on an Apple II running the p-system version known as Apple PASCAL. Members of the original UCSD team can now be found busily writing MODULA-2 compilers or other systems and applications software for many of the leading software houses — all in PASCAL or a derivative.

The most remarkable thing about all this is that it has happened despite the almost complete lack of sponsorship from any major commercial interests. PASCAL succeeds purely on its own merits, and not because any large manufacturers have vested interests in selling it. Our PASCAL programming series will concentrate on the standard language (ISO PASCAL), but some machine specific examples will be given where graphics or operating system calls are required.


The Genealogy of High-Level Imperative Languages
1954-1982
1955
1960
1965
1970
1975
1980
1954-57 FORTRAN (FORmula TRANslator)
The oldest high level language, long favoured by scientists and engineers for numeric and array processing
FORTRAN
1959-60 COBOL (COmmon Business Orientated Language)
This is still the most common language for file handling and large data processing applications
COBOL
1958-60 ALGOL 60 (ALGOrithmic Language)
A general algebraic notation for expressing algorithms. A very influential language, although not widely used itself
ALGOL 60
1963-64 PL/1 (Programming Language 1)
This was an attempt to integrate all three previous languages
PL/1
1963-68 ALGOL 68
A giant and enormously powerful version of ALGOL, defined mathematically by a European committee of computer scientists
ALGOL 68
1968-71 Pascal
A well-structured, small and portable general-purpose language, designed to be powerful and efficient despite its simplicity
Pascal
1969 BCPL (Basic Combined Programming Language)
A very successful (British) language that marries ALGOL's structure with the power and machine-level access of assembler
BCPL
1974-78 C
A systems language derived from BCPL that was used to write the UNIX operating system
C
1975 Concurrent Pascal
A version of Pascal extended for concurrency by Per Brinch Hanson
Concurrent Pascal
1977 UCSD Pascal
An extended version Pascal for systems use
UCSD Pascal
1980-82 ISO Pascal
The internationally standardised definition of Pascal
ISO Pascal
1978-82 Ada (named after Ada, the Countess Lovelace)
An enormous all-purpose language specified by the American Dept. of Defense for all government contracts
Ada
1977-80 Modula-2 (Modular)
Wirth's new modular systems language, finding increasing favour throughout the computing community
Modula-2
1981 Occam (named after William of Occam)
A 'small is beautiful' language for concurrency
Occam
IAN McKINNELL

LOGIC GATE

A device that performs a logical operation is called a *logic gate*. This typically has between two to eight inputs and one or two outputs. Electronic logic gates work on the principle of voltages entering the input channels of the gate and — depending on the function of the gate and the binary pattern on the input ports — the output line will be set either high or low.

The properties of transistors make them ideal devices for the construction of logic gates. A transistor has three lines connected to it — a *collector* and a *base* at one end, and an *emitter* at the other. Because there is a potential barrier between one end of a transistor and the other, current will only be able to pass out of the transistor on the emitter if sufficient voltage is applied to the transistor to enable it to overcome the barrier. In practice, this means that current has to be applied to both the base and the collector lines before current will flow on the emitter line. Thus, both inputs have to be 'on' before the output can be 'on'. Therefore, the transistor can be used as an AND gate.

Using this basic building block in various configurations enables us to construct the other five logic circuits (OR, NOT, NAND, NOR and Exclusive-OR).

LOGIC STATE

Depending on whether the binary signal is either a one or a zero, the *logic state* represents the logical sense of True or False of a given binary signal. Generally, we assume that a binary output of one is a True result. This is known as 'positive logic'. However, this is just a convention. It is also possible, using 'negative logic', to regard binary zero as a True result.

In microelectronics, the binary signal depends on the voltage flowing through the line. Using positive logic as an example, if the output signal from a logic gate is +5v, then the signal is treated as binary one, and the result is considered to be True. Conversely, if the output voltage from the gate is 0v (that is, no current is flowing at all) this is binary zero and taken to mean a False result.

LOGIC SYMBOLS

When designing a system — whether it be a program flowchart, an electronic circuit or a logical operation — it is generally best to use a set of graphic symbols that clearly distinguish between the various operations. It is also helpful if these symbols are universally understood.

When designing a logical system we use the internationally agreed set of *logic symbols*. Depending on whether we are using either positive or negative logic, there are six diagrammatic representations — one representing each of the logical Boolean operations.

Logical operations can also be represented in terms of flip-flops (see page 608) and these also have an agreed set of symbols to represent their functions.

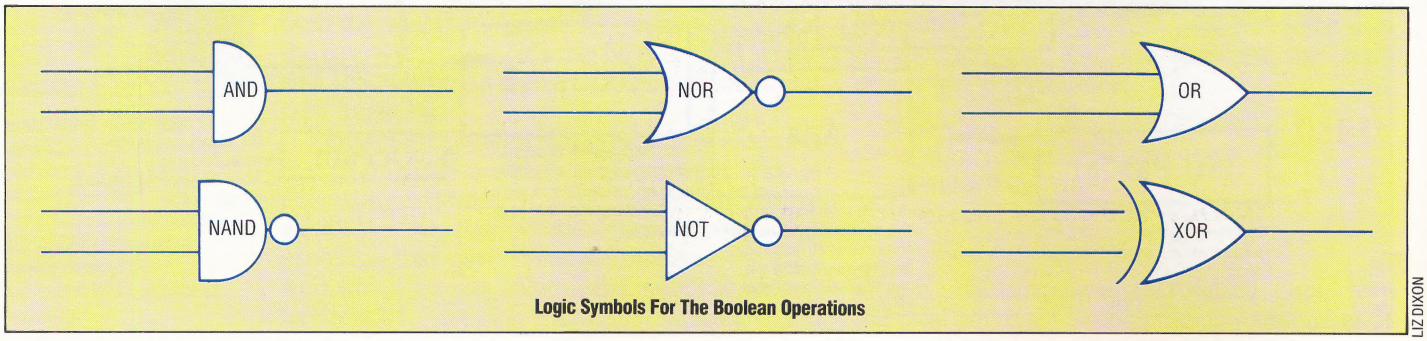
LOGGING ON

When using a microcomputer or a computer terminal it is often necessary to prevent unauthorised persons from gaining access to the computer or certain files. To ensure that only authorised personnel are able to use the system, a process known as *logging on* is built into the system to regulate its use.

Logging on usually involves the use of a password. Often, if a number of people use the same terminal, there will be a common password enabling all the users to gain access to the computer, together with a personal password, allowing the user to access his own personal files, or gain access to parts of the system that are not available to all users.

When finished with the computer, the user *logs off*. This will disconnect the terminal from the computer and delete any files that have not been stored. Many mainframe systems will automatically log off from a terminal if there has been no input for a period of time, enabling other users to log on to the system.

With the growth of computer systems in the past few years, certain computer users, known as 'hackers', have made a hobby of cracking these passwords, allowing them to log on to computers they would otherwise have no access to. In response to this trend, increasingly sophisticated methods of logging on have been developed to deter the hacker. Despite this, however, it appears that hacking is still on the increase. However, one recently developed preventative method may prevent hackers logging on through the telephone network. In this system, if the correct password is given, the computer disconnects the line and then dials back to the number where the password should have originated. If the call originated at a different number, it will not be returned.



Logic Symbols For The Boolean Operations

LIZ DIXON

TO BOLDLY GO

The long-promised Enterprise Sixty Four is an unconventional micro. The machine's 'gamey' looks give the lie to its many well thought out programming facilities and an inspiring capacity for future expansion. Has it arrived too late, though, to make a major impact on the market?

The Enterprise Sixty Four home computer, which was originally to be known as the Elan and then the Flan, set something of a record before it was launched, even in an industry accustomed to long delays between the announcement of a product and its arrival in the shops. The Enterprise took almost two years to reach the shelves.

The problem, as with the Sinclair QL, was that the manufacturer had trouble squeezing the highly impressive specifications onto the custom-built chips. However, the problems have now been solved and the question is, was the wait worth it?

Physically, the Enterprise is certainly unusual, although perhaps not as odd as it promised to be when the machine was announced. Instead of the box-like casing that has become the norm for a home computer, the Enterprise has curved edges around the front of the machine that slope up to the squared-off back. The typewriter keyboard is the same smoky grey as the rest of the casing.

The keys are in all the normal positions. The standard control keys such as Control, Escape and Enter are also sensibly placed. Unusual additions to the control set are two Delete keys — Erase and Delete — which delete backwards and forwards respectively. There is a Hold key, which freezes the running of a program.

The control keys, on either side of the typewriter keys, are green. Above the keyboard are eight blue function keys, which have 16 programmed functions. These follow the fashion of being programmed with commonly used commands such as LIST and AUTO. They are, however, re-programmable.

The red Stop key halts the running of a program, which can be restarted by typing either RUN or CONT. This function is normally carried out by an Escape key. On the Enterprise the Escape key is used to take you from one level of a program up to the next.

The keys themselves are raised slightly above the line of the casing. Although this system is better than the QL's, the moulding and feel of the keys give the impression of being unreliable. Hands-on experience quickly dispels this impression, however, and the keys proved more dependable than some machines with conventional-looking keyboards.

Late Arrival
Although the Enterprise Sixty Four computer was previewed in 1983, it was not until two years later that the machine began to appear on the market in large numbers. The delay has undoubtedly cost the company dearly, as although the machine's specifications were definitely ahead of the field in 1983, the competition has now caught up

The built-in joystick is 3cm ($1\frac{1}{4}$in) long and fitted onto a diaphragm. Games players may find its small size inconvenient for arcade shoot-em-ups.

On the left-hand side of the machine is a cartridge port, called a ROM Bay. Unusually for a modern machine, BASIC is provided on cartridge rather than being resident in the machine — doubtless Enterprise intends other languages and applications to be available on cartridge in the near future. Although the cartridge slot is deeply recessed in the machine, fitting the cartridge gives little trouble. On the opposite side of the machine is a parallel edge connector that will apparently accommodate any future expansion. Enterprise also plans to run the disk drives from the expansion port.

At the rear is a Reset button that provides a warm start to the computer if pressed once. When pressed twice it tells the computer to load a cartridge. This latter facility is necessary as otherwise the computer checks whether a cartridge is fitted only on power up.

Also on the back of the computer are a number of interfaces, including a pair for joysticks and a Centronics-compatible parallel printer port. An RS232/432-compatible serial/net port allows the Enterprise to be connected to a local area network or serial devices and to communicate with other computers via an external modem. A monitor port enables the computer to support an RGB colour monitor.

These interfaces are unusual in that they are all fitted as edge connectors. However, if the machine is successful, third-party suppliers are sure to provide interfaces that allow, for example, Atari-style joysticks to be fitted. The thinking behind this appears to be to allow maximum flexibility in the use of peripherals; Atari-style ports, for example, would be limited to a specific function, whereas an edge connector permits control devices such as touch tablets to be fitted. Nevertheless, when the trend in computers is to 'plug and go', the Enterprise seems out of step with other manufacturers. Of course, this means that, for a while at least, owners will be forced to buy peripherals from Enterprise rather than shop around for them.

Other sockets provided on the back of the machine are a cassette interface with two REMote plugs, which enables the user to run two cassette machines at once — one for input and the other for output. There is also an RF jack for connection to a television and the power socket for the external transformer.

On power-up, the screen displays the Enterprise logo. The user then presses the Enter key and, assuming the BASIC cartridge is not fitted, the built-in word processor is displayed. This is a very useful feature and is greatly superior to the word processor built into the Commodore Plus/4. A number of options are available, selected mostly from the function keys, which enable the user to enter and format the text. The most commonly used functions are displayed at the top of the screen, and others are accessed by pressing f5, which gives you a menu of the other options.

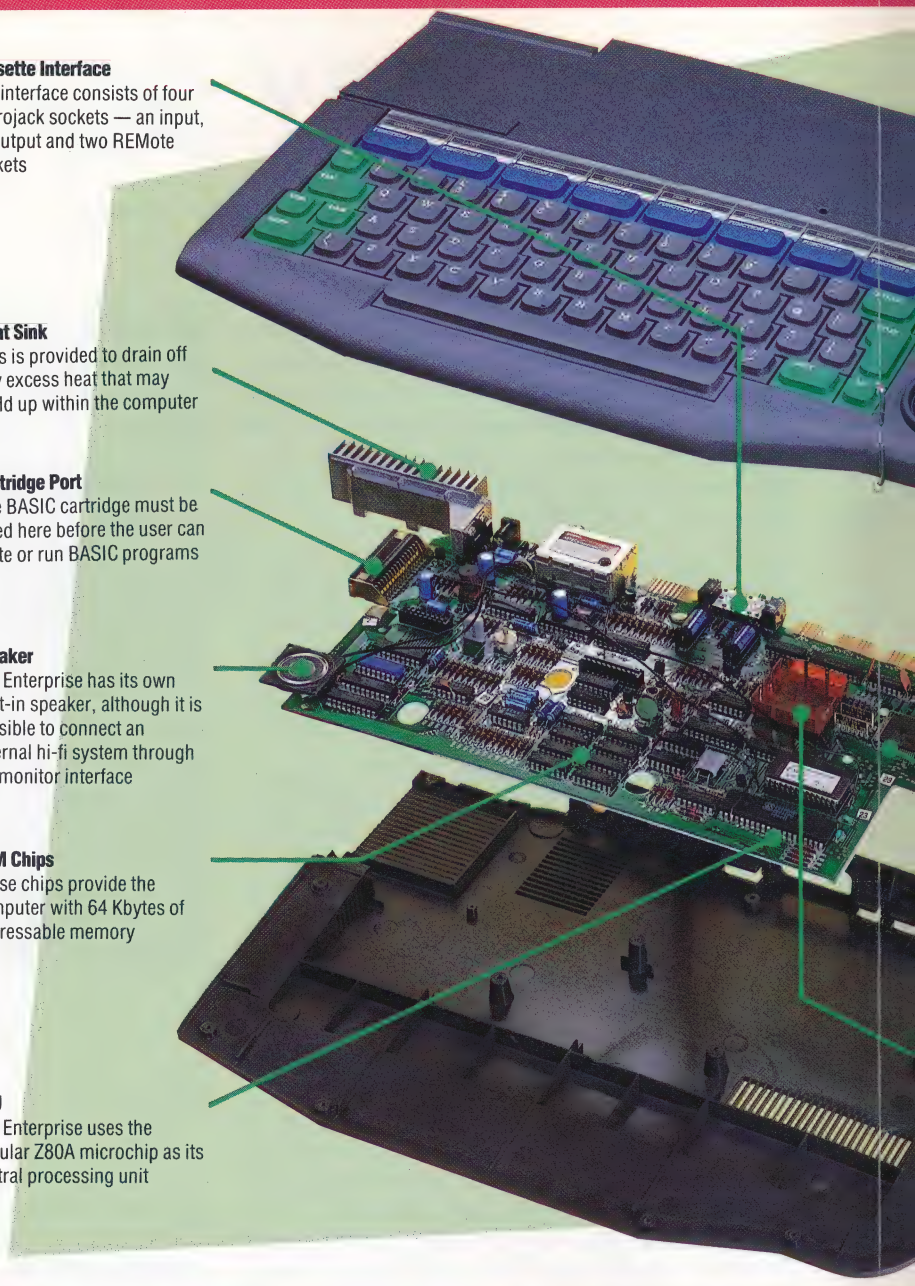
Most surprisingly — and most welcome — is an option that allows the user to write in either 40-column (the default option) or 80-column mode. An 80-column option, combined with the Z80 processor, around which the Enterprise is based, indicates that CP/M will be an early provision on the disk drives when they arrive.

As the Enterprise lacks cursor keys, the cursor is guided around the text screen by means of the joystick. Many users may at first find the joystick difficult to control and the cursor awkward to position precisely, but with a bit of practice the technique of editing with the joystick can be mastered.

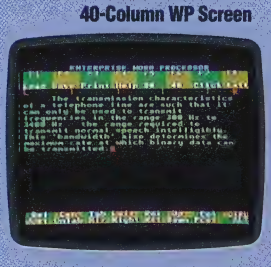
ENTERPRISE BASIC

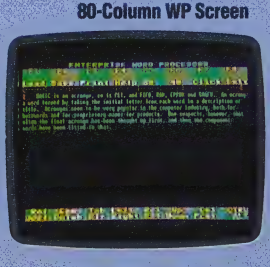
If on power-up the BASIC cartridge is fitted, the screen defaults to BASIC when the Enter key is pressed. Here one sees the advantage of having it held on cartridge rather than resident in the computer. A full 49 Kbytes of memory are available to the user for programming. The BASIC, by Intelligent Software, was one of the early causes of excitement surrounding the Enterprise and certainly seems to be one of the better implementations.

The Enterprise has four sound generators, each of which can be accessed by a SOURCE command. Some of the commands available are PITCH,

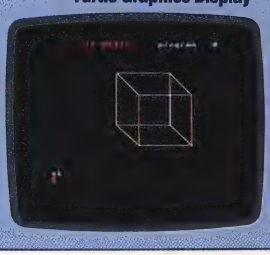

Cassette Interface
The interface consists of four microjack sockets — an input, an output and two REMote sockets
Heat Sink
This is provided to drain off any excess heat that may build up within the computer
Cartridge Port
The BASIC cartridge must be fitted here before the user can write or run BASIC programs
Speaker
The Enterprise has its own built-in speaker, although it is possible to connect an external hi-fi system through the monitor interface
RAM Chips
These chips provide the computer with 64 Kbytes of addressable memory
CPU
The Enterprise uses the popular Z80A microchip as its central processing unit

ROM-Based Software



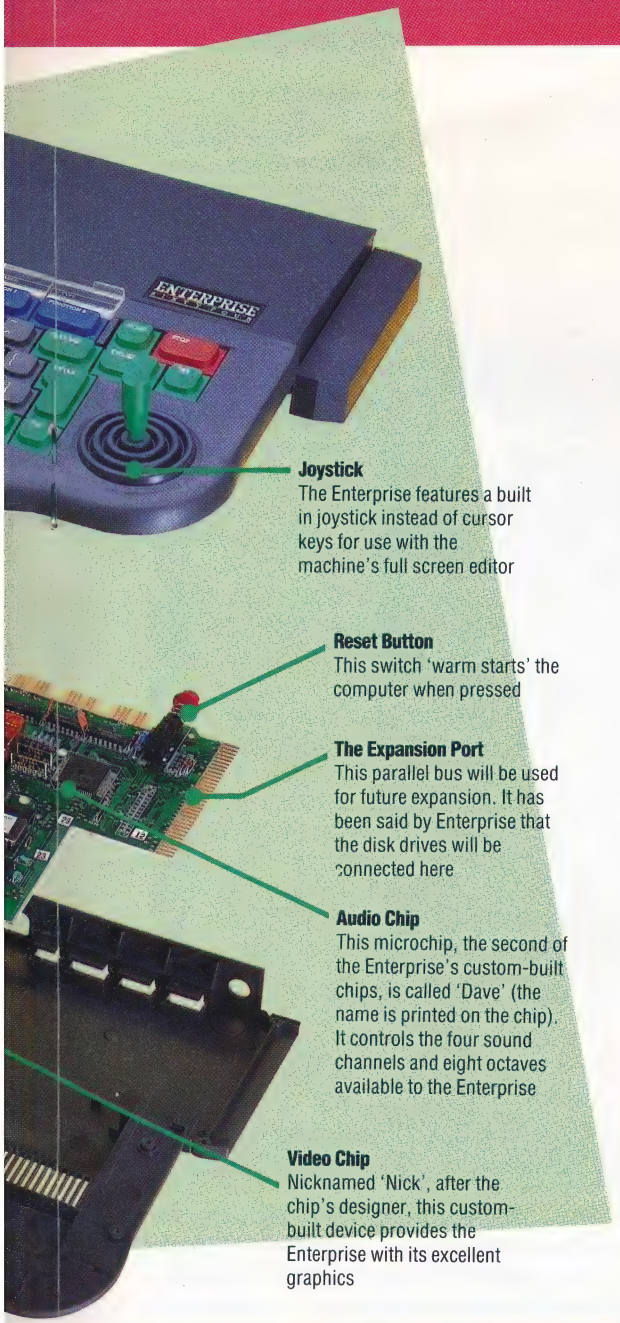


Unlike most other microcomputers, the Enterprise has an on-board word processor. The computer will automatically default to the word processor if there is no cartridge fitted. The word processor can be used in either 40-or 80-column mode. The definition of the text is surprisingly good, even in the 80-column mode



One of the big surprises of the Enterprise is the use of turtle graphics. Apart from the use of the PLOT command, the turtle graphics commands could easily be mistaken for LOGO. By specifying the direction and distance in which the pen is to travel, complex designs can be drawn

IAN McKINNELL



Joystick
The Enterprise features a built in joystick instead of cursor keys for use with the machine's full screen editor

Reset Button
This switch 'warm starts' the computer when pressed

The Expansion Port
This parallel bus will be used for future expansion. It has been said by Enterprise that the disk drives will be connected here

Audio Chip
This microchip, the second of the Enterprise's custom-built chips, is called 'Dave' (the name is printed on the chip). It controls the four sound channels and eight octaves available to the Enterprise

Video Chip
Nicknamed 'Nick', after the chip's designer, this custom-built device provides the Enterprise with its excellent graphics

DURATION and ENVELOPE, enabling the programmer to alter the shape of a note. As the Enterprise can produce stereo sound, the commands LEFT and RIGHT specify from which speaker the sound is to be emitted.

Although the sound produced by the Enterprise chips is excellent, the quality of the built-in speaker is one of the major disappointments of the machine. This is hardly better than that of the Sinclair Spectrum. However, it is possible to connect the computer to a hi-fi system via the monitor edge connector.

The graphics capabilities of the Enterprise are also impressive. Eight graphics modes are available. The user can choose to have either 2, 4, 16 or 256 different colours on screen at the same time in either HIRES or LORES mode. LORES has only half the horizontal resolution of HIRES. The number of colours that may be displayed at any one time also varies according to resolution. Thus, HIRES 2 can produce two colours with a resolution of 640 by 180; in contrast, HIRES 256 can display all 256 colours but has a resolution of only 80 by 180 pixels.

Colours can be chosen by either selecting a number in the range 0 to 255 or, in the case of eight preprogrammed colours, by name — for example, WHITE or RED. Most interestingly, the RGB command enables colours to be selected by mixing them together. Red, for example, corresponds to RGB (1,0,0) and green is RGB (0,1,0). Added together, they produce RGB (1,1,0) — yellow.

If the Enterprise had appeared in 1983 when it was announced, it would have swept the board — the quality of the built-in software alone would have ensured that. The competition is much stronger today, however, and the Enterprise, despite its excellent features, is now just one of a number of alternatives which include new machines from Sinclair and Commodore. Time will tell whether the delays prove fatal to the success of the Enterprise.

ENTERPRISE SIXTY FOUR

PRICE
£249

DIMENSIONS
60×260×405mm

CPU
Z80A running at 4MHz

MEMORY
64K RAM, of which 49K are available for BASIC programs, and 48K ROM (32K for the OS and word processor, and a further 16K available through the BASIC cartridge)

SCREEN
Maximum 80×24 text screen for the word processor. Eight graphics modes available, the highest giving a resolution of 672×512 pixels. Maximum of 256 colours on the screen at any one time

INTERFACES
Centronics-compatible parallel interface, two joystick interfaces, RS232-compatible serial interface, RGB monitor interface, RF port and an expansion bus

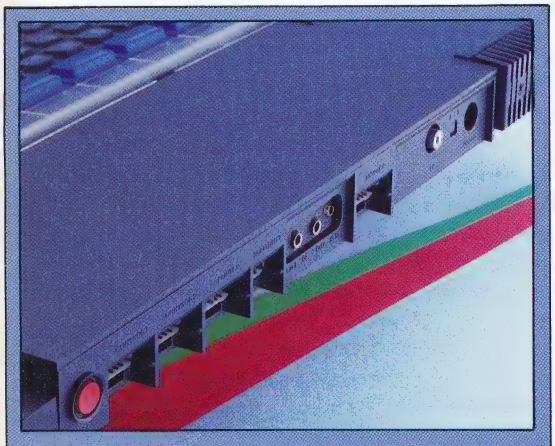
LANGUAGES AVAILABLE
BASIC

KEYBOARD
69 typewriter-style keys, including eight function keys. A joystick is built in

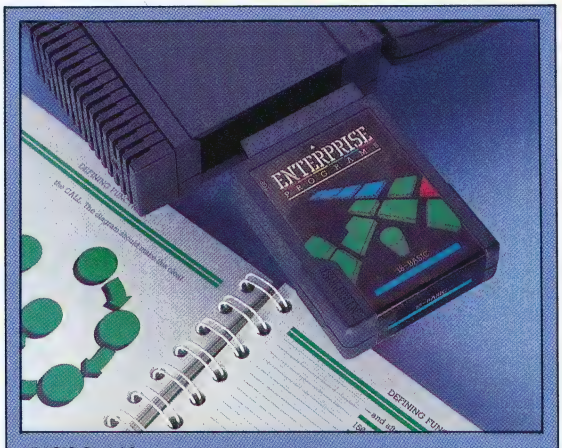
DOCUMENTATION
Included with the computer is a short introductory manual on how to set up the computer, and diagrams explaining the keyboard and interfaces. There is also a BASIC tutorial manual

STRENGTHS
The computer has a good version of BASIC and excellent graphics and sound capabilities, and there's a built-in word processor

WEAKNESSES
In general the hardware is not up to the standard of the software. The keyboard and built-in joystock seem insubstantial. The non-standard interface connection means that, for the time being at least, users will be unable to shop around for peripherals



Peripheral Interfaces
The Enterprise is unusual in having mostly edge connectors as interfaces to peripherals rather than the more usual DIN-type sockets. Although this means that a wider range of peripherals can be connected, in practice many users will probably rely on third party suppliers to provide them with suitable interfaces to the more popular connections



BASIC Cartridge
Unlike most modern machines, the BASIC — by Intelligent Software — is provided on a 16 Kbyte ROM cartridge. This cartridge is fitted into the slot provided on the side of the machine. The provision of a cartridge-based programming language means that other languages can be easily interchanged with BASIC

CHRIS STEVENS

A PLAN TO MODIFY

The unmodified Spectrum lacks the general-purpose user port that enables easy interfacing on both the BBC and Commodore 64 micros. This need not exclude users of the Spectrum from our robot project, however. Here we show how an interface can be designed that will mimic the user port.

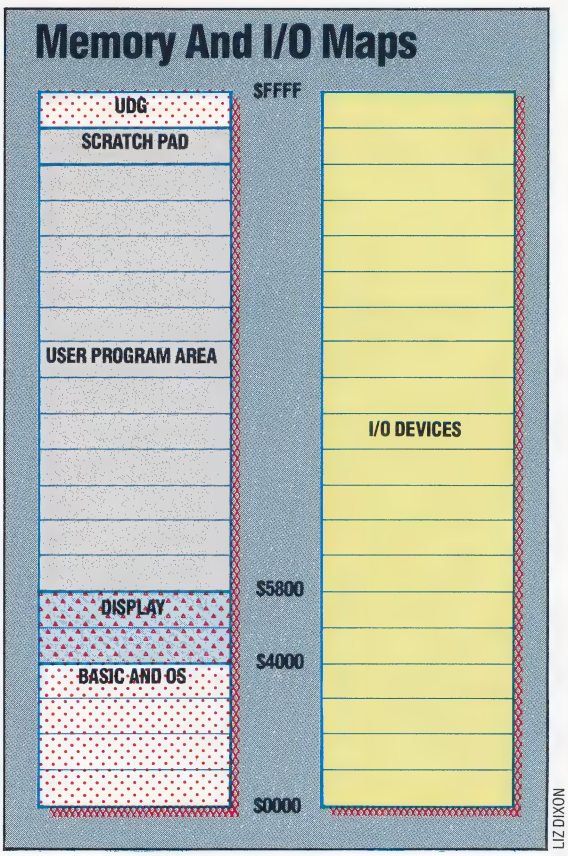
The BBC Micro and Commodore 64 are both equipped with a built-in user port for general-purpose interfacing. The robot that we have built and used in our Workshop course connects to the controlling computer via these interfaces. Unfortunately, the Spectrum has no such user port. In fact, apart from a cassette interface, the Spectrum has no interfaces whatsoever.

Instead, the Spectrum is provided with an expansion port at the back of the machine on which all the internal connections – the data bus, address bus, control lines, and several other signals, including the power lines – are available. These various lines can be used to connect all manner of interfaces to the Spectrum and this is where an interface, to imitate user ports, can be constructed.

The user ports of computers are based around a chip called a VIA, or versatile interface adaptor. This is a complex chip that takes care of a number of tasks. The VIA gives the computer eight input/output lines. Each line (each bit of the I/O byte) can be configured by software as input or output. In addition there are various control input and output lines for 'handshaking' and a built-in timer in the chip. Our DIY Spectrum interface is not so complicated but will fulfil the requirements of the robot. Four output and four input bits are needed. As the in/out configuration is static – it doesn't need to be changed from software – we do not need to imitate the sophisticated DDR/Data Register arrangement of the VIA chip.

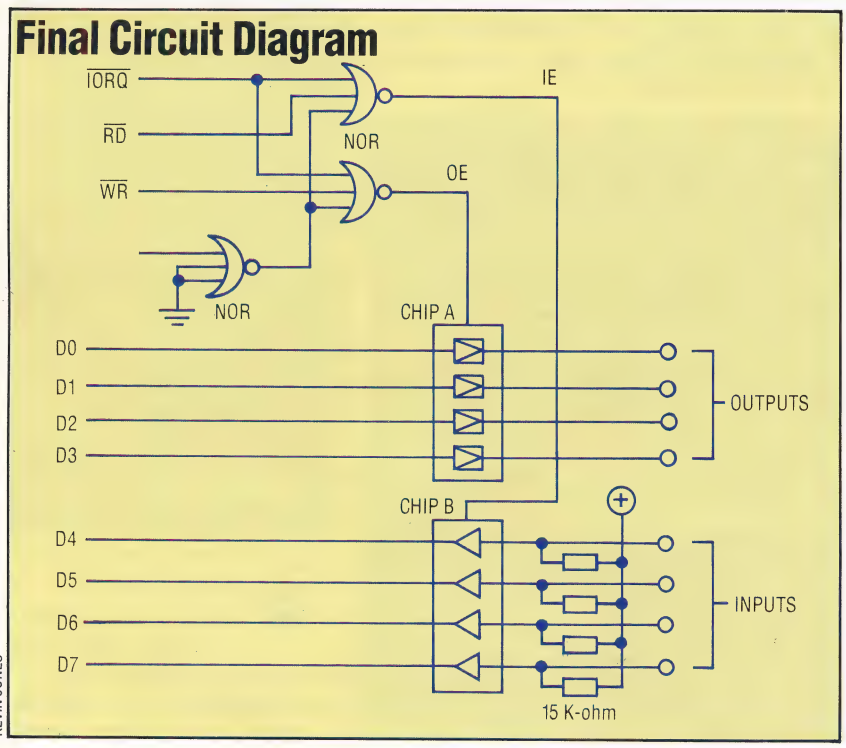
In most computers based around the 6502 microprocessor (such as the BBC Micro and the Commodore 64) the user port and other interfaces are connected into the computer's circuit in the same way as a byte of memory. That is, they form part of the computer's memory map. The Spectrum, however, uses a Z80 microprocessor. This chip has a sophisticated system of interfacing built-in. Instead of mapping I/O devices into the same memory map as the RAM and ROM, a separate memory map is provided for I/O devices. This is in parallel with the main map. It is accessed from BASIC with IN and OUT in the same way as PEEK and POKE are used to

Memory And I/O Maps
The robot is interfaced to the Spectrum by tapping into one of the 65,535 locations in this I/O map — the IN and OUT commands allowing us to read data from, or write data to, the I/O map in the same way that PEEK and POKE are used with the memory map



LIZ DIXON

Logic Lines
By combining the I/O request, read and write, and address bit 5 lines through a series of NOR gates, we can produce two output signals. IE and OE enable the four line input and output buffer chips, allowing control signals to flow out from the computer to the robot's stepper motors — and, conversely, signals from the robot sensors to come in for analysis by software. Input is enabled when $\overline{WR}$ only is high; output being enabled when $\overline{RD}$ only is high



KEVIN JONES

	A10	A8	RFSH	M1	-12v	+12v	WAIT	-5v	WR	RD	IORQ	MREQ	HALT	NMI	INT	D4	D3	D5	D6	D2	D1	D0			D7	A13	A15
A11	A9	BUSACK	ROMCS	A4	A5	A6	A7	RESET	BUSRQ	U	V	Y	VIDEO	0v	IOROGE	A3	A2	A1	A0	CK	0V	0V		9V	5V	A12	A14

access the memory map.

The Z80 distinguishes between the two maps – memory and I/O – with an extra address bit. In addition to the 16 bits that are used to address the 64 Kbytes of memory, another bit controls whether the memory or I/O map is being used. This extra bit is in the form of two control lines from the microprocessor. These are called IORQ (Input/Output ReQuest) and MREQ (Memory REQuest). When a memory location is being addressed, the MREQ line goes low. When an I/O device is to be addressed, the IORQ line goes low. All the 16 address lines and these two control lines are available at the Spectrum's expansion port. We can, therefore, design an interface that will respond only to a certain address and the I/O request.

Some of the 65,535 possible I/O ports are already used in the Spectrum for internal interfacing with the keyboard, cassette interface and beeper. The I/O ports that use address lines 0 to 4 are accounted for by the Spectrum. We will use only address line 5. Even though this means that the interface will 'appear' in many places on the I/O map (at any address using this bit), it simplifies the electronics considerably.

Our interface has to both input and output data so we will need to use the Z80's read (RD) and write (WR) lines as well as the address and control lines. The interface has to respond, then, to the IORQ and A5 lines going low and either the RD or WR line, also going low. When these conditions are satisfied the interface should connect the computer's data bus to the robot's data lines. The truth table for the interface is given.

IE and OE are the enable signals for the input and output parts of the interface, respectively. The final situation of all lines, including both RD and WR, being low at the same time cannot happen (the microprocessor cannot both read and write to a location at the same time). This truth table is most easily implemented in a circuit of the form shown.

When the IORQ and RD and A5 lines all go low (for example, when the microprocessor tries to read from I/O port 31) the IE line in the interface goes high and the four input connections are connected through to the four high bits of the computer's data bus through buffer chip A. When the IORQ and A5 and WR lines go low then OE goes high and the four low bits of the data bus are connected to the four output connections of the interface. When the OE line goes low again, as the processor accesses another address, chip B holds the output pattern until it is again re-accessed, a process known as 'latching'.

If any other combination appears on the four lines, then the interface connections are kept isolated from the data bus and have no effect on it or the computer. This design produces the desired interface for the robot in three chips, one containing the three triple-input OR gates (plus an unused fourth), the quad buffer chip (chip A) and the quad latch (chip B).

Interface Truth Table

IORG	WR	RD	A5	IE	OE
0	0	0	0	—	—
0	0	0	1	—	—
0	0	1	0	0	1
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	0	0	0
0	1	1	1	0	0
1	0	0	0	—	—
1	0	0	1	—	—
1	0	1	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0

Tell The Truth

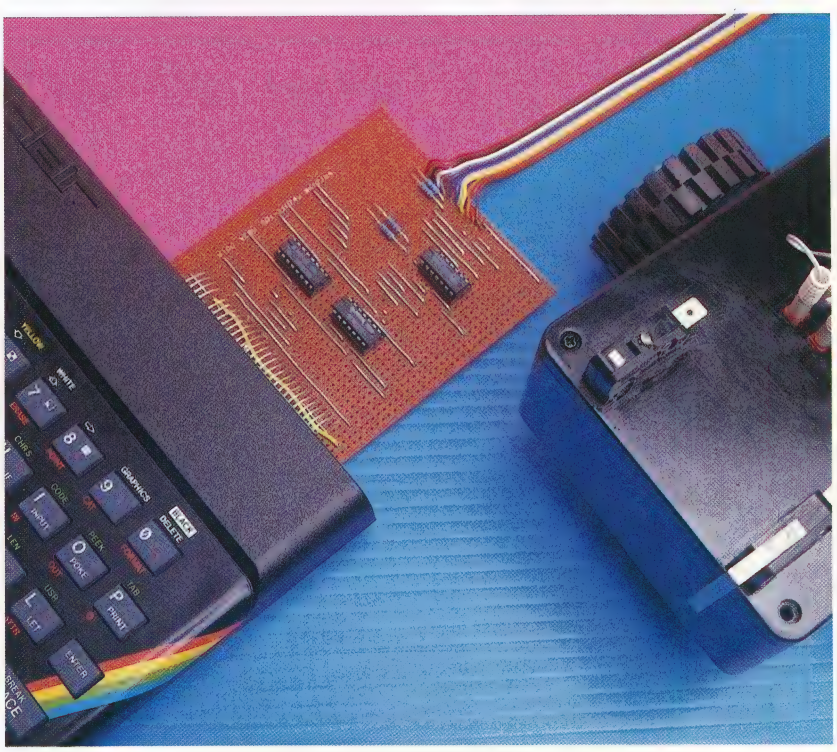
The truth table shows the required signals to be produced on the input and output enable lines, for all possible combinations of the four input signals. The case when WR and RD are both zero cannot happen in practice, since it is impossible to read and write at the same time. We require input to be enabled when IORQ, RD, and A5 are all zero. IE can, therefore, be produced by putting these three lines through a NOR gate. Similarly, output should be enabled when IORQ, WR and A5 are low. A second NOR gate with these three inputs will produce OE

Expanded Horizons

The expansion port on the back of the Spectrum allows external devices to connect with the Spectrum's own internal electronics. The 16 address lines and 8 data lines are available, together with processor pin connections, such as RD, WR and IORQ. By accessing these pins together with the voltage feeds available we can interface our robot to the Spectrum through the expansion port

Interface At Work

The Commodore 64 and BBC Micro have special I/O chips that allow us to connect the Workshop robot directly to the computer. To use the robot with the Spectrum, however, we need to design our own simple interface board that will plug directly into the Spectrum's expansion port. Having built the interface we will be able to control the robot from software



LIZ HEANEY

ON FILE

At this point in the course we introduce the BBC Micro's filing system, which uses a variety of operating system routines to deal with data sent to and accessed from storage media. Here, we examine the OSFILE call in detail, leaving the other filing system ROM routines for the next instalment.

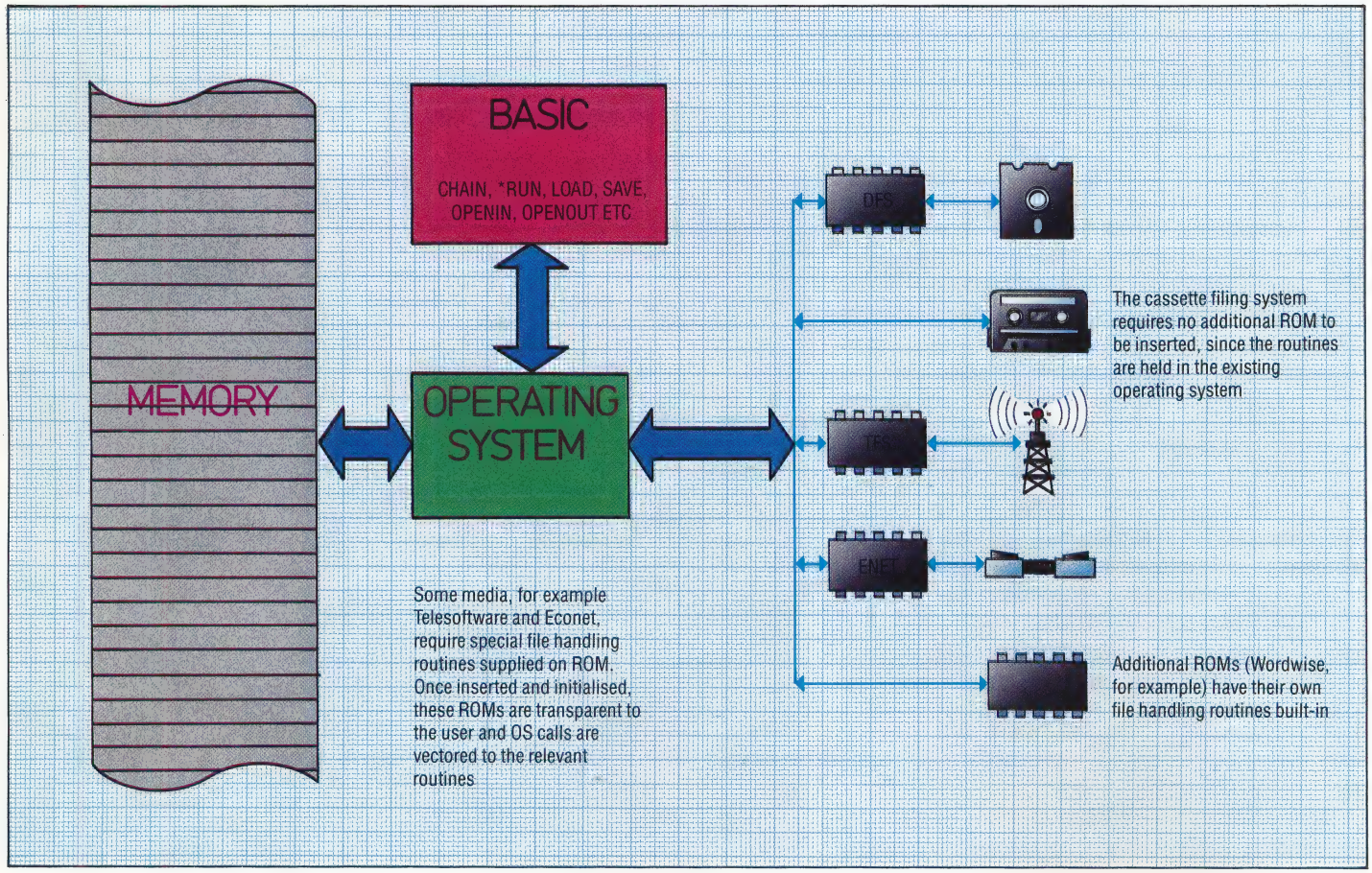
The filing system of a computer deals with the transfer of data between the memory of the computer and a long term storage medium, such as cassette tape or floppy disk. Both of these storage media are supported by the BBC Micro, but other filing devices — such as the Econet network system, ROM cartridges and telesoftware — are also available. Although you may expect there to be different ROM routines, and hence different OS calls, to deal with all these devices, the BBC Micro's filing system uses the same OS routines for all the devices it supports. That is not to say that all devices work in the same way. Separate filing system ROMs are required for each medium to manage the physical file structure (the actual arrangement of the data bytes on the storage medium), but these are interfaced to the logical file structure (records, fields, etc.) through standard OS routines.

The BBC Micro's filing system ROM calls are vectored; when we enter a particular filing system we authorise the OS to set itself up to accommodate the new filing system. Thus, a program that is written using only official OS routines will work under any physical storage system. Of course, some operations — such as a SAVE instruction to a ROM — might not be allowed under a particular filing system.

There are a variety of ways in which a filing system can be entered. We will examine only the most straightforward ways of entering here:

- We can ensure that the filing system is automatically entered on pressing Reset by positioning the appropriate filing system ROM chip in the machine in such a way that it becomes the default ROM chip. Generally, with more than one filing system ROM in the machine, the one to the immediate left of the BASIC ROM will be selected — assuming that BASIC is in the furthest right of the paged ROM sockets.

Filing Cabinet
The filing system provides a set of routines to enable data transfer between memory, long-term storage devices and other media (including Telesoftware and user networks)


BASIC
CHAIN, *RUN, LOAD, SAVE, OPENIN, OPENOUT ETC
MEMORY
OPERATING SYSTEM
DFS
TFS
ENET
The cassette filing system requires no additional ROM to be inserted, since the routines are held in the existing operating system
Some media, for example Telesoftware and Econet, require special file handling routines supplied on ROM. Once inserted and initialised, these ROMs are transparent to the user and OS calls are vectored to the relevant routines
Additional ROMs (Wordwise, for example) have their own file handling routines built-in

KEVIN JONES

- Using BREAK in conjunction with another key. Thus, assuming the appropriate ROM is in the machine, pressing the N key along with BREAK will enter the Econet filing system.
- Using a * command. The command *TAPE 12 will enter the 1200-baud cassette tape system; *DISC enters the disk filing system; *TELESOFT the Telesoftware system and *NET the Econet system. Of course, the latter three commands require the appropriate filing system ROM to be fitted in the machine.

We use the filing system virtually every time we turn the machine on, even if it's just using it to LOAD or SAVE a program. Both these BASIC commands make use of the OS filing system routines — as do CHAIN, OPENUP, OPENOUT, OPENIN, INPUT, PRINT, and so on. A variety of * commands also use OS filing system routines. These are shown in the table. There are other * commands that are provided by individual filing system ROMs.

Before we begin our examination of several specific ROM routines that are involved in filing operations on the BBC, we must point out we don't intend to describe every use of every call that we examine. However, we do describe the main uses of each OS routine.

There are seven OS routines that are concerned with filing only, as well as a couple of OSWORD and OSBYTE calls. The dedicated calls are OSFILE, OSARGS, OSBGET, OSBPUT, OSFSC, OSFIND and OSGBPB. We consider OSFILE in detail here, and will return to the other calls in the next instalment.

THE OSFILE CALL

The OSFILE call operates in terms of complete files of data, which makes it a particularly suitable place to begin. It is used by operations like SAVE, LOAD, *SAVE and so on that need to operate on large blocks of data. These files have pieces of information, called *attributes* associated with them. The attributes of a file include such information as the LOAD ADDRESS, EXECUTION ADDRESS and the file length. OSFILE enables the programmer to manipulate these attributes when it is used with certain filing systems, such as disks. OSFILE, like OSWORD, requires a parameter block — the address of which is passed over to the OSFILE routine in the X and Y registers, just as for OSWORD. OSFILE is also vectored, and we call it at address &FFDD. The actual operation carried out by the OSFILE routine depends upon the contents of the A register when the call is made. We have provided two tables: one showing the structure of the OSFILE parameter block; the other showing the operations performed.

Where operations are being done in a standard BBC Micro, which only requires a 16-bit address to specify memory locations, only the LSB and the next byte are significant in four-byte numbers that specify addresses. If you look at the table of operations performed, it's easy to see that the tape filing system doesn't have many options available to it from OSFILE. It is clearly of greater use for the disk filing system. The file attribute information depends upon the filing system in use. For a disk system, it's very useful, as it enables us to modify, for example, the load address of a file by simply altering the relevant entry in the parameter block and making the OSFILE call with the appropriate value in A.

As an example of the use of OSFILE, the

OSFILE Call Parameter Block

Offset From Address In The X And Y Registers	Description
0	LSB of file name address
1	MSB of file name address
2	LSB of load address of file
3	Not specified
4	Not specified
5	MSB of load address of file
6	LSB of execution address of file
7	Not specified
8	Not specified
9	MSB of execution address of file
10	LSB of data to be written to file for write operation, or destination address for a read
11	Not specified
12	Not specified
13	MSB of data address
14	LSB of (end address of data+1) for a write operation, or file attribute information for read
15	Not specified
16	Not specified
17	MSB of data address

Operation Performed By OSFILE Calls

Contents Of A	Description Of Operation Performed	
	For Disk	For Tape
0	Save area of memory	Save area of memory
1	Write to the catalogue	Load the file
2	Write the load address	Load the file
3	Write execution address	Load the file
4	Write file attributes	Load the file
5	Read the catalogue	Load the file
6	Delete the file	Load the file
255	Load the file	Load the file

following program saves itself using an OSFILE call:

```
10 DIM block 20
20 DIM name 10
30 BASIC=&8023:REM this for BASIC II. &801F for
   BASIC I
40 X%=block MOD 256:REM point to parameter
   block
50 Y%=block DIV 256
60 A%=0:REM specify a write operation
70 block!0=name:REM address of the file name
80 block!2=PAGE:REM load address of program is
90                 :REM going to be PAGE
100 block!6=BASIC:REM execution address will be
110                  :REM the entry point for BASIC
120 block!10=PAGE:REM first byte to be written
130                  :REM is PAGE
140 block!14=(TOP+1):REM end of data + 1
150 $name="FRED":REM put a legal file name in
160 CALL &FFDD:REM do it
170 END
```

Deleting a file from disk with OSFILE is extremely simple — you just put the file name into memory, point the first two entries of the parameter block to the name, set the A register to six and call OSFILE. After a call to OSFILE with A=5, bytes 2 and 5 in the parameter block will hold the load address of the named file, and bytes 6 and 9 will hold the execution address. If the file name found at the address specified in that parameter block cannot be found on the disk, then A will return from the call with the value 200. If the file can be found, then A will hold the value one.

*Commands Using OS Filing System Routines

*Command	Description
*CAT or *.	This generates a catalogue of the files that are currently available on the filing system
*SAVE	This saves a block of bytes from memory to a file medium. For example: *SAVE "filename" start finish execution where 'start' is the address in hexadecimal of the first byte to be saved, 'finish' is the hexadecimal address of the last byte+1 to be saved, and 'execution' is the execution address of the resulting file of bytes. N.B. 'execution' is an optional parameter and if omitted, the execution address is taken to be the same as start; 'finish' can be replaced by '+length', where 'length' is the number of bytes to be saved
*LOAD	Loads a file of bytes to a particular address in the computer memory. The command is used in this form: *LOAD "filename" address where 'address' is the address in hexadecimal where the bytes are to be loaded. If this parameter is omitted, then the bytes are loaded to the address from which they were saved
*RUN	This is like a machine code version of the BASIC CHAIN command. *RUN "filename" will load the file of bytes to the address from which they were saved, and will then execute them as a machine code program, starting at the execution address that was saved from the bytes
*SPOOL and *EXEC	These were discussed in an earlier part of the course (see page 880)
*OPT 1,x	This command instructs the OS whether messages are to be printed during filing systems operations. If x=0, no messages are printed; x=1 gives short messages and x=2 gives longer messages
*OPT 2,x	This command controls the action of the OS when an error occurs during a filing system operation. If x=0 then errors are ignored, x=1 prompts the user to try again when an error is found, and x=3 causes the filing system operation to terminate on finding an error
*OPT 3,x	Sets the time delay between blocks of data being written to tape by the tape filing system. x is in units of 0.1 second
*OPT 4,x	This is used only in disk systems but is included here because of its usefulness. Sets the SHIFT-BREAK options on disk. See DFS manual for details

THE HAUNTED FOREST

We round up our adventure game programming project with a full listing of the Haunted Forest game. In showing how this program was created, we have demonstrated all the basic principles involved in the design of an adventure game. The full listing given here will help you tie together all the component routines we have given.

Haunted Forest is a short adventure game designed to demonstrate some of the basic programming techniques of adventure game programming. Of necessity, the game is limited in both storyline and the number of options open to the player. The construction of Haunted Forest shows how an adventure game program can be built around a simple framework, the basis of which is a map showing all the different locations and their relationships to each other within the adventure world. Whether the game has 16 or 600 locations, it must still be designed using the map as a basis. In addition, objects and special 'perils' can be marked on the map. It is obviously important that the overall storyline and the major perils have been worked out in some detail before programming begins. Often, drawing out the adventure world map on squared paper is the best way to rationalise and form ideas for the rules of the game. Once a final concept has been arrived at, then programming can begin.

Normally the first task is to recode the two-dimensional pictorial map as array elements. In the system used by Haunted Forest, the location data is stored in two one-dimensional arrays, holding a location description and a coded list of the exits available from that location. Objects present in the game are held in a two-dimensional array, indicating a description of the object and the location it resides at. At this stage, we have a simple database representing the adventure world, and simple handling routines can be developed to allow the player to move around the various locations and pick up or drop objects. The specific rules relating to the way the game is played are — at this stage — largely undefined, apart from equipping the game with the objects needed to enable the player to negotiate the hazards presented by the game's special scenarios.

This version of the program was written for the Commodore 64 but will run on most machines that support Microsoft-type BASIC dialects. Minor amendments may be necessary with regard to random number generation and screen clearing. We also give detailed 'Basic Flavours' for the BBC Micro. Because of the Spectrum's peculiar string array handling, however, Spectrum flavours are too numerous to list here. You must refer back to past issues for these. To assist you in collating Spectrum flavours, we do include a conversion table for the string variable names, which are used in all the Spectrum flavours given previously.

Because an adventure game relies on the player 'discovering' perils and hidden procedures in order to negotiate the adventure world, a detailed look at how a game is constructed inevitably 'gives away' the game's scenario. In showing the full internal workings of the Haunted Forest game, therefore, we have of necessity revealed the details of that game's scenario. Our Digitaya game was provided throughout the series in tandem with the Haunted Forest to provide you with an adventure to play without knowing the scenario. We will list Digitaya in the next instalment.

Conversion Table For Spectrum String Variables

Microsoft	Spectrum	Purpose Of Variables
LN$()	L$()	Location description
EX$()	E$()	Exits
IC$()	I$()	Objects held by player
IV$(,)	V$(,)	Objects in game
SN$	S$	Holds sentence for formatting
OW$	O$	'Old word' in format routine
NW$	N$	'New word' in format routine
EX$	X$	Exits from current location
IS$	I$	Current instruction
DR$	D$	Direction specified by instruction
NN$	R$	Non-verb part of instruction
VB$	B$	Verb part of instruction
CD$	C$	Code word

Basic Flavours

BBC Micro:
The following lines should be subtituted in the Haunted Forest listing:

```
207 RND (-TIME)
210 P=RND (10)

1160 CLS

4190 REPEAT: A$=GET$:UNTIL A$="Y" OR A$="N"
4535 CR=RND (3)

6067 CC=0
```

The Final Listing

```
130 REM ** HAUNTED **
140 REM ** FOREST  **
180 :
200 GOSUB6000:REM READ ARRAY DATA
205 GOSUB 1000:REM STORY SO FAR
207 R=RND(-1)
210 P=INT(RND(TI)*10+1):REM START POINT
220 :
230 REM **** MAIN LOOP STARTS HERE ****
240 MF=0:REM MOVE FLAG
245 PRINT
250 GOSUB2000:REM DESCRIBE POSITION
255 GOSUB2300:REM DESCRIBE EXITS
257 GOSUB2700:REM IS P SPECIAL ?
258 IF SF=1 THEN 300:REM NEXT INSTRUCTION
260 PRINT:INPUT"INSTRUCTIONS";IS$
270 GOSUB2500:REM SPLIT INSTRUCTION
275 IF F=0 THEN 260:REM INVALID INSTRUCTION
280 GOSUB3000:REM NORMAL COMMANDS
290 IF VF=0 THENPRINT:PRINT"I DONT UNDERSTAND"
300 IF MF=1 THEN 240:REM NEW LOCATION
310 IF MF=0 THEN 260:REM NEW INSTRUCTION
320 :
990 END
1000 REM **** STORY SO FAR S/R ****
1010 SN$="WELCOME TO THE HAUNTED FOREST"
1020 GOSUB5500:REM FORMAT
1030 PRINT
1040 SN$="AS YOU AWAKE FROM A DEEP SLEEP, THE "
1050 SN$=SN$+"FOREST FLOOR FEELS SOFT AND DRY. "
1060 SN$="YOU DO NOT KNOW HOW YOU CAME TO BE HERE "
1070 SN$=SN$+"BUT KNOW THAT YOU MUST FIND THE "
1080 SN$=SN$+"VILLAGE ON THE EDGE OF THE WOOD TO "
1090 SN$=SN$+"REACH SAFETY."
1100 GOSUB5500:REM FORMAT
1110 PRINT
1120 SN$="YOU LOOK AROUND, TRYING TO GET YOUR BEARINGS."
1130 GOSUB5500:REM FORMAT
1140 PRINT:PRINT"PRESS ANY KEY TO START"
1150 GET A$:IF A$="" THEN 1150
1160 PRINTCHR$(147):REM CLEAR SCREEN
1170 RETURN
1180 :
2000 REM **** DESCRIBE LOCATION ****
2010 SN$="YOU ARE "+LN$(P):GOSUB5500
2020 SN$="YOU SEE "
2030 REM ** CHECK INVENTORY FOR OBJ **
2040 F=0:SP$=""
2050 FOR I=1 TO 3
2060 IF VAL(IV$(I,2))<>P THEN 2080
2070 SN$=SN$+SP$+"A "+IV$(I,1):F=1:SP$=", "
2080 NEXT I
2090 IF F=0 THEN SN$=SN$+"NO OBJECTS"
2100 GOSUB5500:REM FORMAT OUTPUT
2110 RETURN
2120 :
2299 :
2300 REM **** DESCRIBE EXITS S/R ****
2310 EX$=EX$(P)
2320 NR=VAL(LEFT$(EX$,2))
2330 EA=VAL(MID$(EX$,3,2))
2340 SO=VAL(MID$(EX$,5,2))
2350 WE=VAL(RIGHT$(EX$,2))
2353 :
2355 IF(NR OR EA OR SO OR WE)=0 THEN RETURN
2360 PRINT:SN$="EXITS ARE TO THE "
2370 IF NR <>0 THEN SN$=SN$+"NORTH "
2380 IF EA <>0 THEN SN$=SN$+"EAST "
2390 IF SO <>0 THEN SN$=SN$+"SOUTH "
2400 IF WE <>0 THEN SN$=SN$+"WEST "
2410 GOSUB 5500:REM FORMAT
2415 PRINT
2420 RETURN
2430 :
2500 REM **** SPLIT COMMAND S/R ****
2510 IF IS$="LIST" OR IS$="END" THEN VB$=IS$:F=1:RETURN
2515 IF IS$="LOOK" THEN VB$=IS$:F=1:RETURN
2520 F=0
2530 LS=LEN(IS$)
2540 FOR C=1 TO LS
2550 A$=MID$(IS$,C,1)
2560 IF A$<>" " THEN 2590
2570 VB$=LEFT$(IS$,C-1):F=1
2580 NN$=RIGHT$(IS$,LS-C):C=LS
2590 NEXT C
2600 :
2610 IF F=1 THEN RETURN
2620 PRINT:PRINT"I NEED AT LEAST TWO WORDS"
2630 RETURN
2650 :
2700 REM **** IS P SPECIAL S/R ****
2705 SF=0:REM UNSET SPECIAL FLAG
2707 REM ** RANDOM GHOST **
2710 IF P>4 AND RND(1)<0.1 THEN GOSUB 4290:RETURN
2715 :
2716 REM ** OTHER SPECIAL LOCATIONS **
2720 ON P GOSUB4590,4870,5100,4580
2730 RETURN
2735 :
3000 REM **** NORMAL COMMANDS S/R ****
3010 VF=0:REM VERB FLAG
3020 IF VB$="GO" OR VB$="MOVE" THENVF=1:GOSUB3500
3030 IF VB$="TAKE" OR VB$="PICK"THEN VF=1:GOSUB3700
3040 IF VB$="DROP" OR VB$="PUT"THEN VF=1:GOSUB3900
3050 IF VB$="LIST" OR VB$="INVENTORY"THEN VF=1:GOSUB4100
3055 IF VB$="LOOK" THEN VF=1:MF=1:RETURN
3060 IF VB$="END" OR VB$="FINISH" THEN VF=1:GOSUB4170
3070 RETURN
3080 :
3500 REM **** MOVE S/R ****
3505 GOSUB3630:REM SEARCH FOR DIRECTION
3510 MF=1:REM SET MOVE FLAG
3520 DR$=LEFT$(NN$,1)
3530 IF DR$<>"N"ANDDR$<>"E"ANDDR$<>"S"ANDDR$<>"W"THENGOTO3590
3540 IF DR$="N"AND NR<>0 THEN P=NR:RETURN
3550 IF DR$="E"AND EA<>0 THEN P=EA:RETURN
3560 IF DR$="S"AND SO<>0 THEN P=SO:RETURN
3570 IF DR$="W"AND WE<>0 THEN P=WE:RETURN
3580 PRINT:PRINT"YOU CAN'T "IS$
3585 MF=0:RETURN
3590 REM ** NOUN NOT DIRECTION **
3600 PRINT"WHAT IS "NN$" ?"
3610 MF=0:RETURN
3620 :
3630 REM **** SEARCH FOR DIRECTION S/R ****
3640 NN$=NN$+" ":LN=LEN(NN$):C=1
3645 FOR I=1 TO LN
3650 IF MID$(NN$,I,1)<>" " THEN NEXT I:RETURN
3655 W$=MID$(NN$,C,I-C):C=I+1
3660 IF W$="NORTH" OR W$="EAST" THEN NN$=W$:I=LN
3665 IF W$="SOUTH" OR W$="WEST" THEN NN$=W$:I=LN
3670 NEXT I
3675 RETURN
3700 REM **** TAKE S/R ****
3710 GOSUB 5300:REM IS OBJECT VALID
3720 IF F=0 THEN SN$="THERE IS NO "+W$:GOSUB5500:RETURN
3730 OV=F:GOSUB5450:REM CHECK INVENTORY
3740 IF HF=1 THEN SN$="YOU ALREADY HAVE THE "+IV$(F,1):GOSUB5500:RETURN
3750 :
3755 REM ** IS OBJECT HERE ? **
3760 IF VAL(IV$(F,2))<>P THEN SN$=IV$(F,1)+" IS NOT HERE":GOSUB5500:RETURN
3770 :
3780 REM ** ADD OBJECT TO LIST **
3790 A=0
3800 FOR J=1 TO 2
3810 IF IC$(J)="" THEN IC$(J)=IV$(F,1):AF=1:J=2
3820 NEXT J
3830 :
3840 REM ** FULL QUOTA **
3850 IF AF=0 THEN PRINT"YOU ALREADY HAVE TWO OBJECTS":RETURN
3860 :
3870 SN$="YOU TAKE THE "+IV$(F,1):GOSUB5500
3880 IV$(F,2)="-1":REM DELETE INVENTORY ENTRY
3890 RETURN
3895 :
3900 REM **** DROP S/R ****
3910 GOSUB5300:REM VALID OBJECT
3920 IF F=0 THEN SN$="THERE IS NO "+W$:GOSUB5500:RETURN
3930 :
3940 REM ** IS OBJECT IN CARRIED INVENTORY **
3950 OV=F:GOSUB5450
3960 IF HF=0 THEN SN$="YOU DO NOT HAVE THE "+IV$(F,1):GOSUB5500:RETURN
3970 :
3980 REM ** DROP OBJECT **
3990 SN$="YOU DROP THE "+IV$(F,1):GOSUB5500
4000 IV$(F,2)=STR$(P):REM MAKE ENTRY IN INVENTORY
4010 :
4020 REM ** DELETE OBJECT FROM CARRIED INVENTORY **
4030 FOR J=1TO2
4040 IF IC$(J)=IV$(F,1) THEN IC$(J)="":J=2
4050 NEXT J
4060 RETURN
4070 :
4100 REM **** LIST CARRIED INVENTORY ****
4110 PRINT"OBJECTS HELD:"
4120 FOR I=1 TO 2
4130 PRINT"  ";IC$(I)
4140 NEXT I
4150 RETURN
4160 :
4170 REM **** END GAME S/R ****
4180 PRINT:PRINT"ARE YOU SURE (Y/N) ?"
4190 GET A$:IF A$<>"Y" AND A$<>"N" THEN 4180
4200 IF A$="N" THEN RETURN
4210 END
4220 :
4290 REM **** RANDOM GHOST S/R ****
4295 SF=1:GC=0
4300 SN$="YOU FEEL A COLD SENSATION RUNNING THE LENGTH"
4305 SN$=SN$+" OF YOUR SPINE. SUDDENLY A WHITE APPARITION"
4310 SN$=SN$+" APPEARS FROM OUT OF THE TREES AND"
4315 SN$=SN$+" MOVES TOWARDS YOU":GOSUB5500:REM FORMAT
4320 :
4325 SN$="THE GHOST MOVES CLOSER":GOSUB5500
4330 GC=GC+1:IF GC>4 THEN GOSUB4455:REM
4335 PRINT:INPUT"INSTRUCTIONS";IS$
4340 GOSUB2500:REM SPLIT INSTRUCTION
4345 IF F=0 THEN 4325:REM NEXT INSTRUCTION
4350 OP=P:GOSUB3000:REM ANALYSE INSTRUCTION
4355 IF MF=1 AND VB$="GO"THEN GOSUB4400:GOTO 4325
4357 IF MF=1 AND VB$="LOOK" THEN GOSUB2000:GOSUB2300:GOTO4325
4360 IF VF=1 THEN 4325:REM NEXT INSTRUCTION
4365 REM ** NEW INSTRUCTION WORDS **
4370 IF VB$="KILL" OR VB$="FIGHT" THEN GOSUB4425:GOTO 4325
4375 :
4385 IF VB$="SING" THEN GOSUB4500:RETURN
4390 SN$="I DON'T UNDERSTAND":GOSUB5500:GOTO4325
4395 :
4400 REM ** ATTEMPT TO MOVE **
4405 SN$="YOU ARE TRANSFIXED WITH TERROR AND CANNOT"
4410 SN$=SN$+" MOVE...YET":MF=0:GOSUB5500:P=OP
4415 RETURN
4420 :
4425 REM ** FIGHT OR KILL **
4430 SN$="THE GHOST IS A BEING OF THE SUPERNATURAL,"
4435 SN$=SN$+" AND LAUGHS AT YOUR FEEBLE ATTEMPTS"
4440 SN$=SN$+" TO INJURE HIM":GOSUB5500
4445 RETURN
4450 :
4455 REM ** DEATH **
4460 SN$="THE PAIN IN YOUR CHEST BECOMES UNBEARABLE"
4465 SN$=SN$+" AND YOU SLUMP ONTO THE LEAFY FOREST FLOOR.":GOSUB5500
```

KEVIN JONES

```
4470 SN$="YOUR SPIRIT RISES FROM YOUR INERT BODY"
4475 SN$=SN$+" AND YOU FLOAT AWAY INTO THE MIST TO JOIN"
4480 SN$=SN$+" THE OTHER TORMENTED SOULS OF THE"
4485 SN$=SN$+" HAUNTED FOREST.":GOSUB5500
4490 END
4495 :
4500 REM ** SING **
4505 SN$="YOU KNOW THREE SONGS. WHICH ONE WILL YOU CHOOSE ?":GOSUB5500
4510 SN$="1) THE THEME FROM 'GHOSTBUSTERS'":GOSUB5500
4515 SN$="2) 'THERE'S A GHOST IN MY HOUSE'":GOSUB5500
4520 SN$="3) 'WAY DOWN UPON THE SWANEE RIVER'":GOSUB5500
4525 PRINT:INPUT"MAKE YOUR CHOICE";C$
4530 IF VAL(C$)>3 OR VAL(C$)<1 THEN PRINT:PRINT"INVALID":GOTO4525
4535 CR=INT(RND(1)*3)+1
4537 IF CR<>VAL(C$) THEN GOSUB4542:REM WRONG TUNE
4540 GOSUB4565:RETURN:REM CORRECT
4542 REM **** WRONG TUNE S/R ****
4545 SN$="THE GHOST HAS A PARTICULAR HATRED OF"
4550 SN$=SN$+" THAT TUNE AND LUNGES AT YOU.":GOSUB5500
4555 GOSUB 4455:REM DEATH
4560 :
4565 REM ** CORRECT TUNE **
4570 SN$="THE GHOST IS APPEASED BY YOUR RENDITION OF THE TUNE"
4575 SN$=SN$+" AND VAPOURISES INTO THIN AIR":GOSUB5500
4580 RETURN
4585 :
4590 REM **** TUNNEL ENTRANCE S/R ****
4600 SF=1
4605 SN$="YOU HAVE ARRIVED AT THE MOUTH OF A LARGE TUNNEL":GOSUB5500
4610 SN$="YOU CAN ENTER THE TUNNEL OR RETREAT ALONG THE PATH":GOSUB5500
4620 :
4625 PRINT:INPUT"INSTRUCTIONS";IS$
4630 GOSUB2500:REM SPLIT INSTRUCTION
4635 IF F=0 THEN 4625:REM INVALID INSTRUCTION
4637 GOSUB3000:REM NORMAL INSTRUCTIONS
4640 IF MF=1 THEN RETURN:REM PLAYER RETREATS
4645 IF VF=1 THEN 4625:REM INSTRUCTION OBEYED
4650 REM ** NEW INSTRUCTIONS **
4655 IF VB$="ENTER" THEN GOSUB 4700:RETURN
4660 IF VB$="RETREAT" AND P=4 THEN MF=1:P=6:RETURN
4665 IF VB$="RETREAT" AND P=1 THEN MF=1:P=9:RETURN
4667 SN$="I DON'T UNDERSTAND":GOSUB5500:GOTO 4625
4700 REM ** ENTER TUNNEL **
4705 SN$="YOU ENTER THE TUNNEL BUT IT IS TOO DARK TO"
4710 SN$=SN$+" FIND YOUR WAY.":GOSUB5500
4725 PRINT:INPUT"INSTRUCTIONS";IS$
4730 GOSUB2500:REM SPLIT INSTRUCTION
4732 :
4735 IF F=0 THEN 4725:REM INVALID INSTRUCTION
4740 OP=P:GOSUB3000:REM NORMAL INSTRUCTIONS
4745 IF MF=1THEN SN$="IT IS SO DARK THAT YOU CAN ONLY SEE":P=OP
4747 IF MF=1THENSN$=SN$+" THE TUNNEL ENTRANCE":GOSUB5500:MF=0:GOTO4725
4750 IF VF=1 THEN 4725:REM INSTRUCTION OBEYED
4755 IF VB$="RETREAT" AND P=4 THEN MF=1:P=6:RETURN
4760 IF VB$="RETREAT" AND P=1 THEN MF=1:P=9:RETURN
4762 IFVB$()"USE"ANDVB$()"LIGHT"THEN SN$="I DON'T UNDERSTAND"
4765 IFVB$()"USE"ANDVB$()"LIGHT"THEN GOSUB5500:GOTO4725
4777 :
4780 REM ** SEARCH FOR LAMP **
4790 GOSUB5300:REM VALID OBJECT ?
4795 OV=F:GOSUB5450:REM IS OBJECT HELD ?
4797 IF F=0 THEN SN$="THERE IS NO "+W$:GOSUB5500:GOTO4725
4800 IF HF=0 THEN SN$="YOU DO NOT HAVE THE "+IV$(F,1):GOSUB5500:GOTO4725
4810 REM ** IS OBJECT LAMP ? **
4815 IF F<>2 THEN SN$="THE "+IV$(F,1)+" IS NO USE":GOSUB5500:GOTO4725
4835 REM ** SUCCESS **
4840 SN$="YOU USE THE LAMP TO LIGHT YOUR WAY THROUGH THE TUNNEL"
4845 SN$=SN$+" AND EVENTUALLY EMERGE FROM THE EXIT.":GOSUB5500
4850 IF P=1 THEN MF=1:P=4:RETURN
4855 IF P=4 THEN MF=1:P=1:RETURN
4860 :
4870 REM **** SWAMP S/R ****
4875 SF=1
4880 SN$="YOU START TO SINK INTO THE SWAMP.":GOSUB5500
4885 PRINT:INPUT"INSTRUCTIONS";IS$
4890 GOSUB2500:REM SPLIT INSTRUCTION
4895 IF F=0 THEN 4885:REM INVALID
4900 GOSUB3000:REM NORMAL INSTRUCTIONS
4910 IF VB$="LOOK" THENGOSUB2000:GOTO4885
4915 IF VB$="DROP"THEN IV$(F,2)="-2":REM OBJ LOST FOREVER
4917 IF VF=1 THEN 4885:REM NORMAL COMMAND
4920 REM ** NEW COMMANDS **
4925 IF VB$()"SWIM"THEN SN$="I DON'T UNDERSTAND":GOSUB5500:GOTO4885
4930 REM ** SWIM **
4932 F=0
4935 FOR I=1 TO 2
4940 IF IC$(I)<>""THEN F=F+1
4950 NEXT I
4955 IF F<2 THENGOSUB5035:RETURN:REM SWIM AWAY
4960 GOSUB 5000:RETURN:REM TWO OBJS HELD
5000 REM **** TWO OBJECTS HELD S/R ****
5010 SN$="THE OBJECTS ARE WEIGHING YOU DOWN AND YOU ARE SINKING.":GOSUB5500
5012 PRINT:INPUT"INSTRUCTIONS";IS$
5015 GOSUB2500:REM SPLIT INSTRUCTION
5020 IF VB$()"DROP" THENGOSUB5080:REM SINK
5025 GOSUB3900:IV$(F,2)="-2":REM DROP OBJ
5030 IF HF=0 OR F=0 THEN 5080:REM SINK
5035 REM **** SWIM AWAY ****
5040 SN$="YOU CAN NOW SWIM THROUGH THE SWAMP. WHICH WAY WILL YOU GO?"
     :GOSUB5500
5050 EX$(2)="00080605":GOSUB2300:REM DEFINE AND DISPLAY EXITS
5055 PRINT:INPUT"INSTRUCTIONS";IS$
5060 GOSUB2500:REM SPLIT INSTRUCTION
5062 IF F=0 THEN 5055:REM INVALID
5065 GOSUB3500:REM MOVE
5067 EX$(2)="00000000":REM ZERO EXIT DATA
5070 RETURN
5075 :
5080 REM **** SINK S/R ****
5085 SN$="YOU SINK INTO THE SWAMP AND DROWN":GOSUB5500
5090 END
5100 REM **** VILLAGE S/R ****
5102 SF=1
5105 SN$="THE VILLAGE IS SURROUNDED BY A TALL WALL.":GOSUB5500
5106 IF GF<>0 THEN GOSUB5190:RETURN:REM GATE
5107 SN$="A GUARD IS AT THE ENTRANCE GATE TO THE VILLAGE":GOSUB5500
5115 PRINT:INPUT"INSTRUCTIONS";IS$
5120 GOSUB2500:IF F=0 THEN 5115:REM INVALID
5125 GOSUB3000:REM NORMAL INSTRUCTIONS
5130 IF VB$="LOOK"THEN GOSUB2000:REM DESCRIBE
5135 IF VB$="GO" AND MF=1 THEN RETURN
5140 IF VF=1 THEN 5115:REM NEXT INSTRUCTION
5145 IF VB$<>"KILL"THENSN$="I DON'T UNDERSTAND":GOSUB5500:GOTO5115
5150 REM ** KILL **
5155 SN$="WHAT WILL YOU USE TO KILL THE GUARD?":GOSUB5500
5160 SN$="ENTER OBJECT OR <I> FOR INSTRUCTION":GOSUB5500
5162 INPUT IS$:IF IS$="I" THEN 5115
5165 GOSUB2500:REM SPLIT
5167 IF F=0 THEN 5160:REM INVALID
5170 GOSUB5300:IF F=0 THEN SN$="THERE IS NO "+W$:GOSUB5500:GOTO5160
5172 OV=F:GOSUB5450:REM IS OBJECT HELD
5174 IF HF=0 THEN SN$="YOU DO NOT HAVE THE "+IV$(F,1):GOSUB5500:GOTO5160
5175 IF F<>1 THEN SN$="THE "+IV$(F,1)+" IS NO USE":GOSUB5500:GOTO5160
5180 SN$="YOU KILL THE GUARD":GOSUB5500:GF=1
5185 :
5190 REM **** LOCKED GATE S/R ****
5195 SN$="YOU MOVE FORWARD AND TRY TO OPEN THE GATE TO THE VILLAGE"
5200 SN$=SN$+" BUT THE GATE IS LOCKED AND WILL NOT MOVE":GOSUB5500
5205 PRINT:INPUT"INSTRUCTIONS";IS$
5210 GOSUB2500:IF F=0 THEN 5205:REM INVALID
5215 GOSUB3000:REM NORMAL INSTRUCTIONS
5220 IF VB$="LOOK"THEN GOSUB2000:REM DESCRIBE
5225 IF VB$="GO" AND MF=1 THEN RETURN
5230 IF VF=1 THEN 5205:REM NEXT INSTRUCTION
5232 IF VB$="USE"THEN 5240
5234 IF VB$="UNLOCK"THENSN$="HOW?":GOSUB5500:GOTO5205
5235 SN$="I DON'T UNDERSTAND":GOSUB5500:GOTO5205
5240 GOSUB5300:REM VALID OBJECT
5242 OV=F:GOSUB5450:REM IS OBJ CARRIED
5244 IF F=0 THEN SN$="THERE IS NO "+W$:GOSUB5500:GOTO5205
5246 IF HF=0 THEN SN$="YOU DO NOT HAVE THE "IV$(F,1):GOSUB5500:GOTO5205
5248 IF F<>3 THEN SN$="THE "+IV$(F,1)+" IS NO USE":GOSUB5500:GOTO5205
5250 REM ** THROUGH GATE AND SAFE **
5255 SN$="YOU UNLOCK THE GATE, AND DISGUISING YOURSELF IN THE DEAD"
5260 SN$=SN$+" GUARD'S CLOTHES, WALK UNNOTICED THROUGH THE VILLAGE"
5265 SN$=SN$+" AND THE SAFETY OF THE OUTSIDE WORLD.":GOSUB5500
5270 END
5299 :
5300 REM **** VALID OBJECT S/R ****
5310 NN$=NN$+" ":LN=LEN(NN$):C=1:F=0
5315 FOR K=1 TO LN
5320 IF MID$(NN$,K,1)<>" " THEN NEXT K:RETURN
5325 W$=MID$(NN$,C,K-C):C=K+1
5330 LW=LEN(W$)
5335 FOR J=1 TO 3
5340 LI=LEN(IV$(J,1)):REM LENGTH OF OBJECT
5350 FOR I=1 TO LI-LW+1
5360 IF MID$(IV$(J,1),I,LW)=W$ THEN F=J:I=LI:J=3:K=LN
5370 NEXT I,J,K
5380 RETURN
5390 :
5450 REM **** IS OBJECT HELD S/R ****
5460 HF=0
5470 IF IV$(OV,2)="-1" THEN HF=1
5480 RETURN
5490 :
5500 REM **** FORMAT OUTPUT S/R ****
5510 LC=0:    REM CHAR/LINE COUNTER
5520 OC=1:    REM OLD COUNT INITIAL VALUE
5530 OW$="":  REM OLD WORD  INITIAL VALUE
5540 LL=40:   REM LINE LENGTH
5550 SN$=SN$+" DUMMY "
5560 PRINT
5570 FOR C=1 TO LEN(SN$)
5580 LC=LC+1
5590 IF MID$(SN$,C,1)=" " THEN GOSUB5600
5600 NEXT C
5605 PRINT
5610 RETURN
5620 :
5800 REM ** END OF LINE CHECK S/R **
5810 NW$=MID$(SN$,OC,C-OC+1):REM NEW WORD
5820 IF LC<LL THENPRINTOW$;:GOTO5840
5830 PRINTOW$:LC=LEN(NW$)
5840 OC=C+1:OW$=NW$
5850 RETURN
6000 REM **** READ OBJ & MAP DATA ****
6010 DIM IV$(3,2),LN$(10),EX$(10),IC$(2)
6020 FOR C=1 TO 3
6030 READ IV$(C,1),IV$(C,2)
6040 NEXT C
6050 :
6060 FOR C=1 TO 10
6065 READ LN$(C),EX$(C)
6070 CC=CC+VAL(EX$(C)):REM CHECKSUM TOTAL
6080 NEXT C
6090 :
6100 READ CD:IFCD<>CC THENPRINT"CHECKSUM ERROR":STOP
6110 :
6120 REM ** OBJECT DATA **
6130 DATA GUN,10,LAMP,9,KEY,5
6140 :
6150 REM ** MAP DATA **
6160 DATA NEAR A TUNNEL ENTRANCE,00000800
6170 DATA IN A SWAMP,00000000
6180 DATA IN A VILLAGE,07000000
6190 DATA NEAR A TUNNEL ENTRANCE,05060000
6200 DATA ON A PATH,00020400
6210 DATA ON A PATH,02070004
6220 DATA ON A PATH,00000306
6230 DATA ON A PATH,09000702
6240 DATA ON A PATH,01100800
6250 DATA IN A CLEARING,00000008
6260 REM ** CHECKSUM DATA **
6270 DATA 32253121
6280 RETURN
```

WHO YOU GONNA CALL?

The home computer games market has now become so competitive that software companies are turning to other media to supply them with themes. Here we look at Ghostbusters — a stimulating computer game from America derived from the film of the same name.

Ghostbusters has so far proved a phenomenally popular film in both the USA and the UK. In launching the computer game of the same name, the publishers, Activision, doubtless expected the magic to work once again.

The package was developed in conjunction with the film's makers, Columbia Pictures. This co-operation meant that the programmers were able to follow the film script closely without the fear of being sued by the copyright owners. The game also uses the film's theme tune, which became a best-selling single in the summer of 1984.

Ghostbusters is the story of three college researchers who set up an agency in New York to catch ghosts, and discover that poltergeists are converging on an apartment building in preparation for the end of the world. The game begins with the player being given cash to spend on a car and 'ghost busting' weapons. These include a 'PK energy detector', to warn of approaching ghosts, a 'ghost vacuum', to suck up the spirits, and a portable laser confinement system, which enables the player to drive the ghosts into a special trap. Once the weapons have been purchased, the player must venture out onto the streets of New York to try to earn a living by trapping ghosts.

The scene then changes to a map of the city. At the centre of the map is the building on which the ghosts are converging, attracted by the evil Zuul. The spooks are displayed at various locations around the city and the player must drive the Ghostbuster car to these locations. As the car travels along the city streets, ghosts called 'Roamers' will appear on the screen. If the player has equipped the car with a 'ghost vacuum', which is fitted on the roof, the Roamers can be sucked up to score extra points.

If the ghost is still there when the car has arrived at the location, you can set up the ghostbusting equipment. A trap is first placed at the centre of the screen, and the ghost can then be driven towards this with the lasers held by the ghostbusters. When the ghost is directly above, the trap can be sprung and the ghost will be drawn into the trap. The Ghostbuster theme tune, which plays in the background throughout, will pause at this point and a voice will congratulate you with the salutation 'Ghostbuster!'.

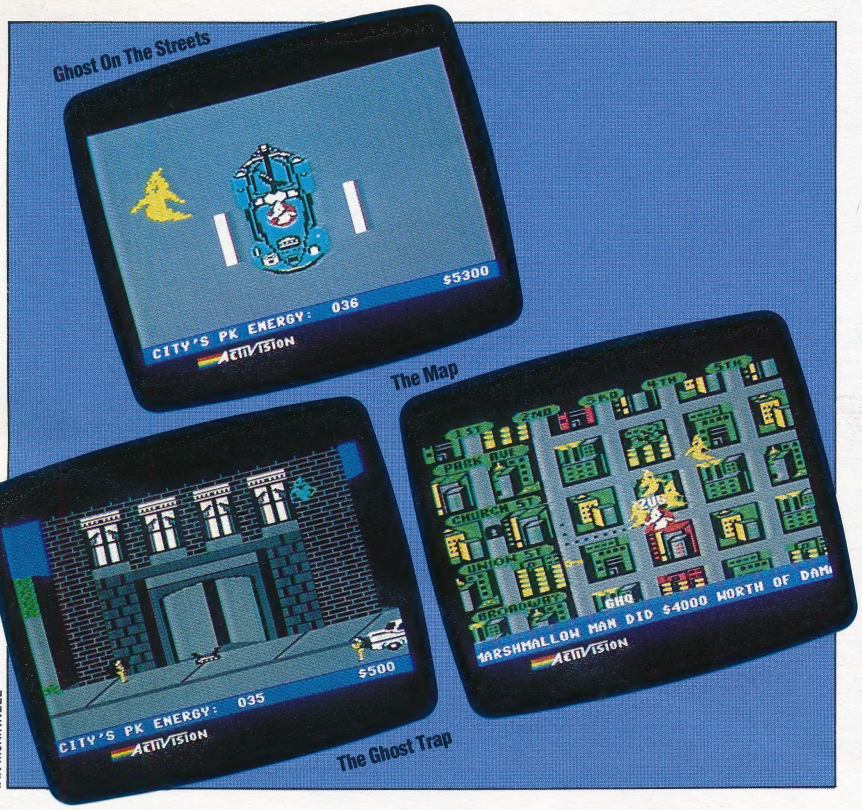
While all this is going on, the player must take care to watch for the Marshmallow alert. This means that the Roamers are joining together to form the Marshmallow Man who will destroy the city unless some ghost bait is dropped immediately. The player must also watch to see whether the Gatekeeper and the Keymaster are about to join forces at the Temple of Zuul. If this happens before the player has amassed enough points to get the ghostbusters into the Temple, then all is lost.

Ghostbusters is a well crafted game, with finely detailed graphics and a catchy tune. The speech output is an added bonus. The game is highly addictive and even the most jaded games players have been known to spend hours chasing the elusive spectres.

Ghostbusters: For the Commodore 64 £10.99, Sinclair Spectrum £9.99
Publishers: Activision (UK) Inc, 15 Harley House, Marylebone Road, Regents Park, London NW1
Authors: David Crane, Adam Bellin, Hilary Mills.
Joysticks: Required
Format: Cassette

Ghosts In The Machine
Shown here are three screen shots of the game Ghostbusters by Activision. While travelling to the scene of a 'haunting', ghosts will pass you on the streets. These can be captured, provided you have fitted a ghost trap to the top of your car.

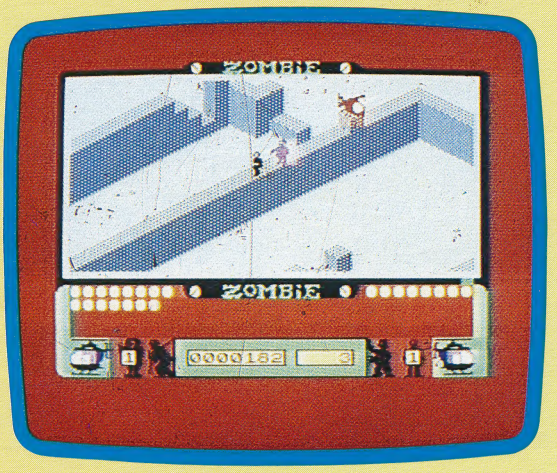
Once at the scene of the haunting, the ghosts are driven towards a trap with laser beams. The map shows the location of the Temple of Zuul, and indicates the presence of the Marshmallow Man who wanders around destroying parts of the city



IAN McKINNELL

In the past, THE HOME COMPUTER ADVANCED COURSE has reviewed only that software which we regard as highly innovative or imaginative – a 'classic' in other words. But there is an awful lot of good software around – especially games programs – which we simply haven't had the space to assess. Here, we present the first in a series of reviews of software written for the more popular home micros. This week, our reviewers are Ross Holman, Dave Nicholls and Roger Willis; and the games are for the Sinclair Spectrum.

JOYSTICK JURY



ZOMBIE ZOMBIE

Quicksilva, £6.95

Dave Nicholls: Oh no! Antescher has been invaded by zombies – nasty green ones that turn red with rage and attack whenever you get too close.

To help you rid the city of this influx of *Duran Duran* fans, you're equipped with a helicopter which you can use to fly around in safety. But you have to leave it when you're destroying the zombies – a feat accomplished by running into them from behind! Once hit they turn all slavelike and can be led up onto any convenient wall where they will happily jump to their death to the tune of *Ten Green Bottles*. If you're not too confident of controlling the zombies this way, you can always fire your 'puffer' at them which'll make them run away to a safe distance

Any architects out there will also be glad to hear that your helicopter is capable of carrying a large number of bricks around, so the city can be re-defined to suit yourself; you can also have some fun with your own brand of 'Soft Solid graffiti'!

Basically this is *Ant Attack* all over again and as such it will appeal to Ant Attackers everywhere. But if you found *Ant Attack* a tedious experience then don't bother with *Zombie Zombie*. For those of you who've seen neither, however, *Zombie Zombie* is probably the best of the two. *3/5* HIT

Ross Holman: This is just too similar to *Ant Attack* to offer anything very exciting. The tunes are OK and the ability to write graffiti on the bricks should keep you amused. *1/5* MISS

Roger Willis: Tidying up the undead can be amusing, even though I'm told it's a pastime short on originality. Probably a hit. *3/5* HIT

EDDIE KIDD JUMP CHALLENGE

Martech, £6.95

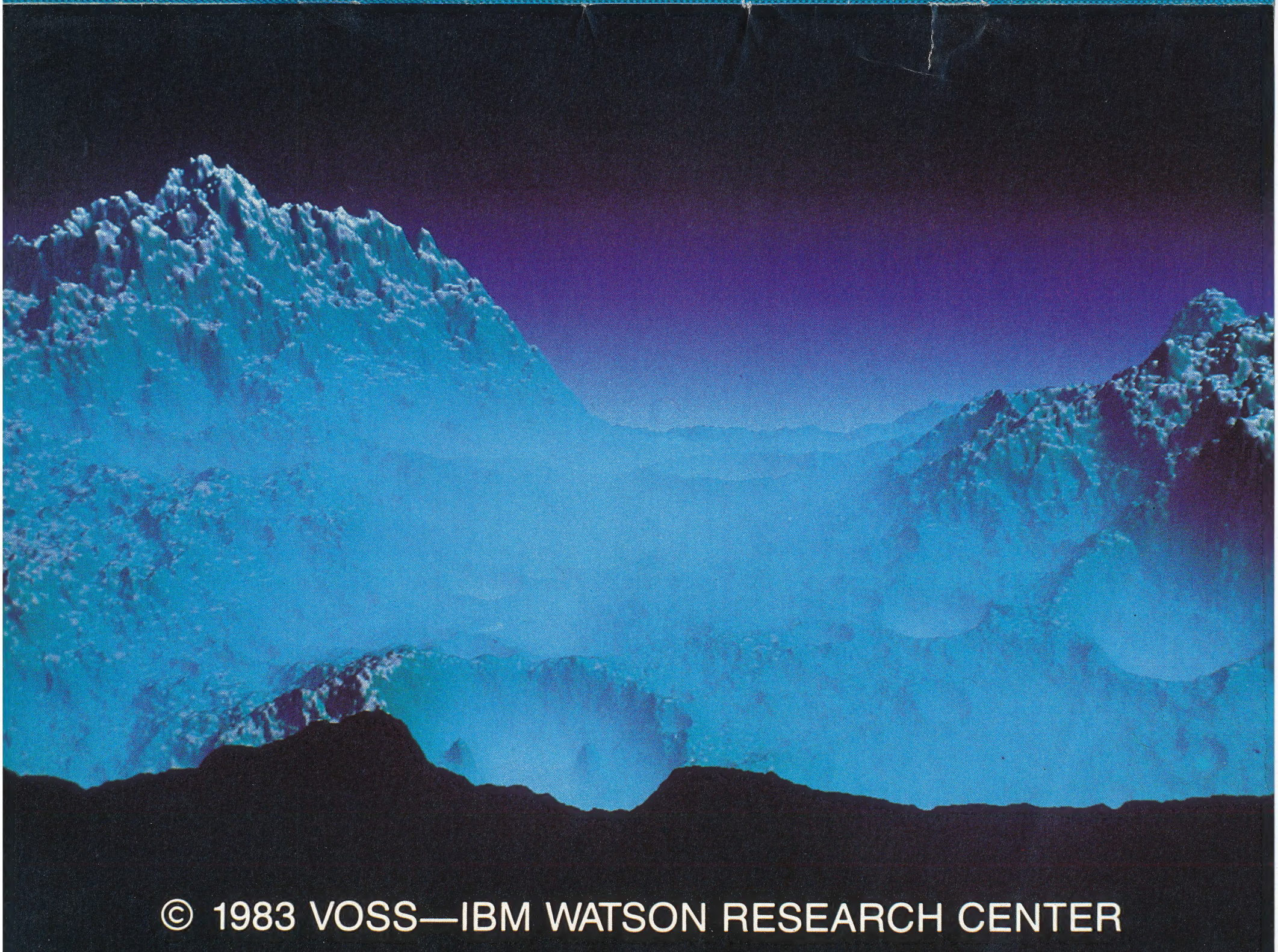
Roger Willis: I can tell you from personal experience that leaning back or forward can certainly affect your stability while jumping a motorcycle. I can also tell you that, just as in this program, getting it wrong has spectacular consequences. Here, though, Eddie always gets up and waves to the crowd – I usually stay down screaming with pain...

Approach distance appears irrelevant, so long as the bike has climbed through its five gears and indicated speed is at its maximum when taking off from the ramp. All you really have to do is control the bike in the air and that, presumably, governs the number of cars successfully jumped over. Sounds fine, doesn't it? Well, it isn't...

Cheapskate graphics only serves to underline the fact that Eddie Kidd has attached his name to a fairly primitive arcade game that gets boring quicker than players will get good at it. The intro – or practice – screen demands you jump a pushbike over oildrums; trouble is, you make a perfect jump every time! Oh yes, jumps beyond a certain limit produce an offer to SAVE on to blank cassette for competition entry purposes, but I can't imagine anyone bothering. *1/5* MISS

Dave Nicholls: The BMX section gets annoying – it's so easy you can do it with your eyes closed! Jumping with cars is much more difficult and quite addictive. *4/5* HIT

Ross Holman: You can control Eddie quite well, but even after practise I found it difficult to make any successful jumps over a decent distance. The 'gusting wind' didn't help either! *2/5* MISS


© 1983 VOSS—IBM WATSON RESEARCH CENTER